UNIVERSIDAD CARLOS III DE MADRID

ESCUELA POLITÉCNICA SUPERIOR

INGENIERÍA DE TELECOMUNICACIÓN

PROYECTO FIN DE CARRERA

Laboratorio de malware: Automatización de la gestión de recursos virtuales para el estudio de malware

Autor: Antonio Parra Truyol

Tutor: Guillermo Suárez de Tangil

Co-Tutor: Sergio Pastrana Portillo

Mayo 2013

HOJA EN BLANCO

A mi hermana

RESUMEN

En la actualidad, la seguridad en los sistemas de información se considera una prioridad tanto para el entorno empresarial como para el personal. Cada día hay más información crítica y, aunque los sistemas presentan cada vez una mayor robustez, también su complejidad supone un reto a la hora de identificar vulnerabilidades en los sistemas.

Teniendo en cuenta estos aspectos, es necesario contar con los medios necesarios para poder facilitar el estudio de malware y anticipar cuáles son los sistemas afectados por el mismo y bajo qué circunstancias. De esta forma se podrá mitigar el riesgo derivado del mal uso de los sistemas.

En este proyecto se han diseñado unos laboratorios donde poder realizar un estudio del *malware*. En los mismos se podrán ejecutar experimentos en entornos aislados que servirán para poder analizar el comportamiento del *malware*. Se proporciona también la habilidad de crear una infinidad de entornos de red con diferentes sistemas operativos y aplicaciones que facilitarán el estudio del mismo. Para crear estos entornos aislados, se ha hecho uso de la virtualización de sistemas. Esta tecnología va a permitir crear entornos aislados, flexibles y escalables donde poder ejecutar cuantos experimentos sean necesarios. Por último, se realizan una serie de experimentos y se muestran los resultados. Como resultado de los experimentos, se puede comprobar que la creación de distintos entornos resulta útil para poner de manifiesto distintos comportamientos maliciosos en función del *malware* ejecutado y de las aplicaciones instaladas sobre los distintos sistemas operativos.

Palabras clave: *Malware*, seguridad informática, vulnerabilidad, virtualización, *cloud computing*, hipervisores y sistemas SIEM.

ABSTRACT

Currently, system security is considered a priority for both business and personal environment. Every day, there is more and more critical information in the systems and, although systems are more robust, they are also more complex posing a challenge to vulnerability discovery.

Considering these aspects, it is paramount to anticipate to malware and ensure the security of the systems. This will ensure data integrity and system robustness.

In this project we have designed a laboratory of malware. In the laboratory, experiments can be executed in isolated environments, and they will be used to dissect malware behavior in an automatic fashion. More precisely, the laboratory provides the ability to create an infinite number of network environments combining different operating systems and applications that facilitates the study of malware in the wild. To create these isolated environments, we have used virtualization technology. This technology will allow creating isolated environments, flexible and scalable, where we can run as many experiments as needed. Finally, we conduct some experiments and results are shown. Based on our experiments, we conclude that the different environments are enough to manifest malicious behaviors and we describe differences between malware executions depending on each environmental configuration.

Keywords: Malware, security information, vulnerability, virtualization, cloud computing, hypervisors and SIEM systems.

Tabla de contenidos

Índice de figuras

Índice de tablas

Capítulo 1

1. INTRODUCCIÓN Y OBJETIVOS

1.1. Introducción

Actualmente, la tecnología avanza a un ritmo espectacular. Apenas se está terminando de completar el estudio y análisis de una tecnología, cuando ya se ha desarrollado otra más moderna que la desborda y sustituye. Este cambio es frenético y difícil de controlar: aparecen actualizaciones de sistemas cada mes, semana, e incluso, se puede hablar de cuestión de días.

Cualquier sistema, por muy complejo y sofisticado que sea, se encuentra expuesto, de forma creciente, a un peligro que se ha convertido en uno de los principales problemas en el diseño de sistemas hoy en día: el "MALWARE". La existencia de una pieza de *malware* viene determinada, en gran medida, por las repercusiones económicas que puede generar su actividad. Es por esta razón que el *malware* es principalmente una amenaza para:

- Sistemas operativos populares como pueden ser los sistemas operativos Windows debido al gran impacto que pueden ocasionar.
- Sistemas específicos con un elevado valor añadido como pueden ser los sistemas de una empresa para la competencia o las infraestructuras críticas de un país para un enemigo.

Malware (del inglés *malicious* software o *badware*), también conocido como software malicioso o software malintencionado, constituye un tipo de software cuyo fin principal es infiltrarse en un sistema, no sólo sin el consentimiento de su propietario, sino en contra de los intereses del mismo y, generalmente, buscando el beneficio, profesional o económico, de quien lo emite. Para ello, el *malware* busca explotar cualquier vulnerabilidad, cosa que es difícil evitar completamente debido a su propia complejidad, que presente el sistema objetivo. Por ello, el *malware* constituye una grave preocupación para cualquier institución, empresa o particular, dado que la información sensible, personal o incluso accesoria se encuentra expuesta al continuo ataque del *malware*.

Paralelo al ritmo de la expansión de los sistemas, el desarrollo del *malware* no se ha quedado atrás. La escalada del número de nuevos programas maliciosos existentes, ha experimentado una línea de crecimiento exponencial como se puede apreciar a continuación en la *Figura 1. Evolución del número de piezas maliciosas nuevas a lo largo de los años.* :

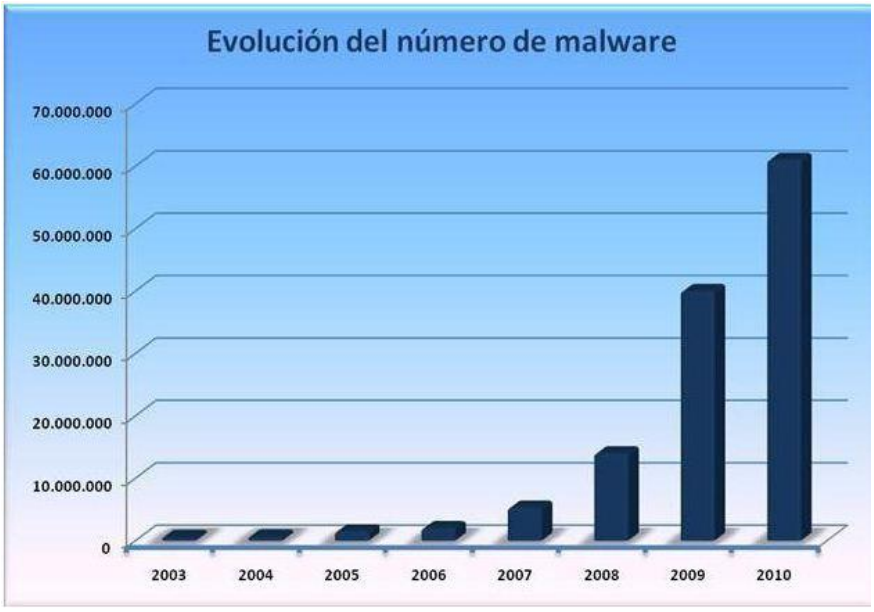


Figura 1. Evolución del número de piezas maliciosas nuevas a lo largo de los años. [1]

El software se considera *malware* en función de los efectos que su creador pretende generar, pero avanzando más en su definición, ésta puede también clasificarse en función de los tipos de efectos que produce, soliendo incluirse dentro de su definición los conceptos conocidos como virus, gusanos, troyanos y otro tipo de software malicioso e indeseable. Sin embargo, la mayor parte de las infecciones las realizan los denominados *troyanos*, como se puede apreciar en la *Figura 2. Infecciones existentes por tipo de Malware, segundo trimestre 2012* .[2] sacada de un reciente estudio[2] sobre *malware*.

[1] http://www.pandasecurity.com/spain/

[2] http://prensa.pandasecurity.com/wp-content/uploads/2012/08/Informe-Trimestral-PandaLabs-Abril-Junio-2012.pdf

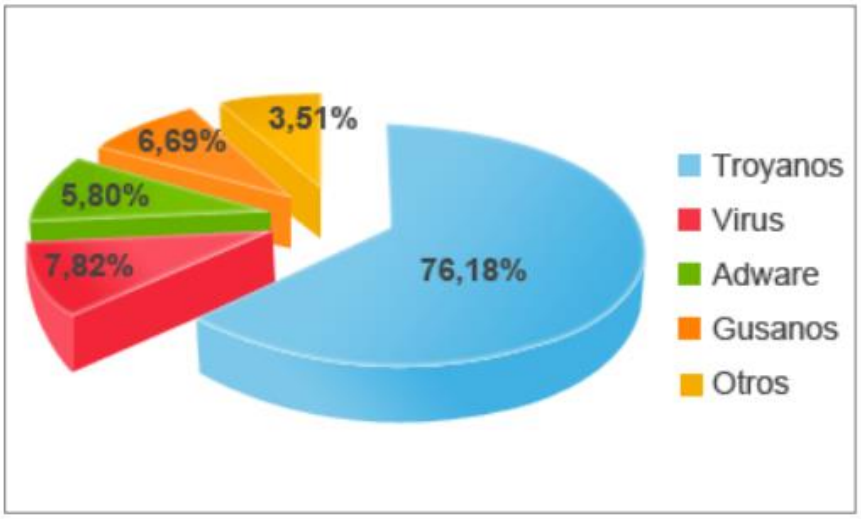


Figura 2. Infecciones existentes por tipo de Malware, segundo trimestre 2012 .[2]

De forma resumida, los *troyanos* constituyen "un software malicioso que se presenta al usuario como un programa aparentemente legítimo e inofensivo pero al ejecutarlo ocasiona daños"[3]. Su funcionamiento consiste principalmente en la toma del control de un sistema para realizar acciones sobre él sin necesidad de permisos por parte del propietario. En este sentido, además de ser los *troyanos* el tipo de *malware* que más predomina en número de infecciones, también son los que experimentan un crecimiento relativo mayor en el tiempo como se refleja en el siguiente gráfico:

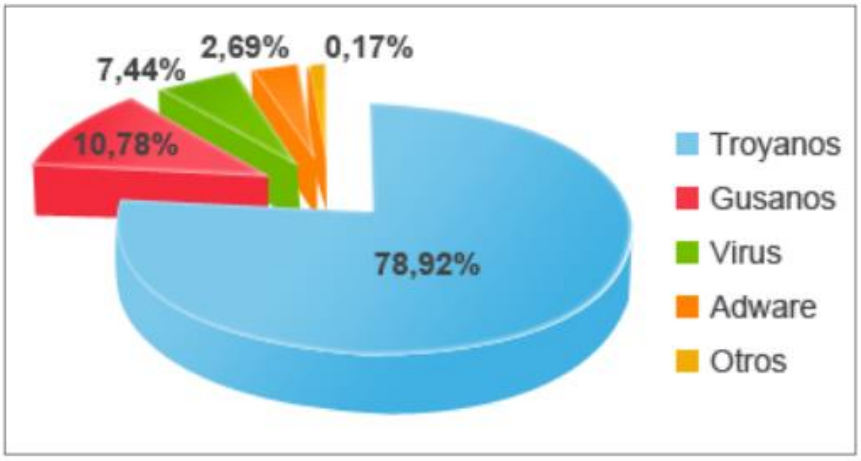


Figura 3. Nuevo malware creado por tipo.[2]

Como consecuencia de los riesgos para la sociedad de la información implícitos en la evolución del *malware*, tanto cuantitativos como cualitativos, este proyecto realiza un estudio exhaustivo del mismo. Para ello se utilizará la técnica de la virtualización de sistemas, que consiste esencialmente en crear un entorno hardware compartido por varios sistemas operativos. Se adopta la decisión de crear un entorno virtual para la creación del laboratorio debido, principalmente, a razones de seguridad, ya que en teoría es posible crear un entorno aislado de ejecución, aunque en la práctica es conveniente

[3] http://es.wikipedia.org/wiki/Troyano_(inform%C3%A1tica)

ser prudentes. Adicionalmente, la construcción del entorno virtual supone un ahorro de costes, tanto en términos de tiempo, como en términos económicos.

La virtualización[4] constituye una tecnología perfectamente válida para poder estudiar el comportamiento del *malware*, en un entorno aislado. Al igual que sucede con el *malware*, la evolución de la misma ha experimentado asimismo un crecimiento exponencial durante los últimos años. Hace poco tiempo, la virtualización era una gran desconocida para casi todos nosotros y aunque se comenzaba a escuchar la palabra *virtual*, todavía no se podía ni siquiera intuir el trasfondo que llegaría a alcanzar. Actualmente en el entorno de las Tecnologías de la Información y la Comunicación, comúnmente conocidas como las TIC, se trata la *virtualización* como un concepto básico del día a día, un concepto que empieza a formar parte del lenguaje coloquial y que cada vez experimenta una mayor difusión en cuanto a su conocimiento o su significado, no sólo por parte de los propios expertos sino por el público en general.

En la *Figura 4. Evolución de la virtualización.* se puede observar cómo existe una tendencia a *hospedarse* los sistemas cada vez más en un mundo virtual:

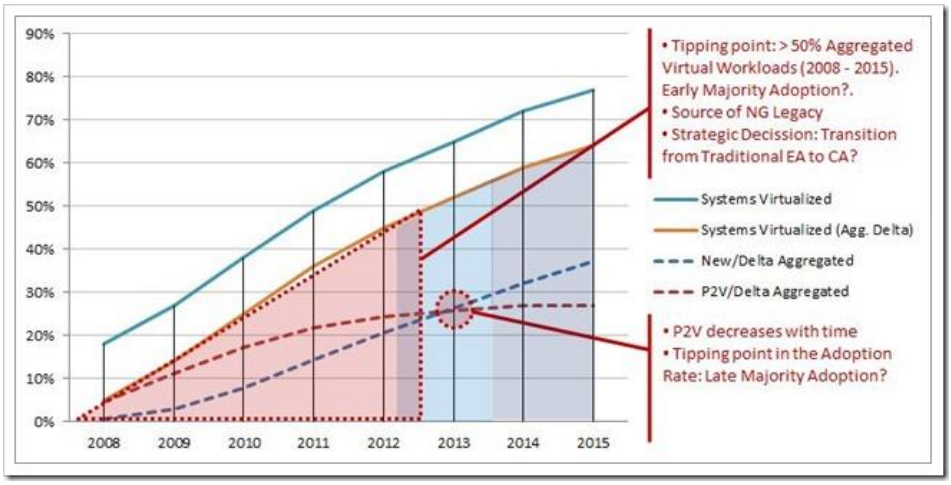


Figura 4. Evolución de la virtualización. [5]

La causa de este enorme desarrollo es evidente, y se fundamenta en los enormes beneficios aportados por esta tecnología, beneficios que se irán desgranando a lo largo del proyecto. Puede adelantarse que uno de los principales beneficios lo constituye el ahorro económico que reporta a las empresas la introducción de esta nueva tecnología. Un ahorro importante que se obtiene, por otra parte, con unos costes de implementación cada vez menores, como se deduce de la *Figura 5. Comparativa del coste de una aplicación instalada en un entorno virtual y no virtual*. en el que se aprecia claramente la disminución del coste de la implantación de un aplicativo al instalarse en un entorno virtual, en relación al coste de instalarse en un entorno no virtual.

[4] *Virtualización* es la tecnología que permite tener varias instancias de sistemas operativos en el mismo equipo, aislados entre sí y del sistema operativo base.
[5] http://thinkinbig.org/index.php/2011/05/thinking-about-cloud-strategy/

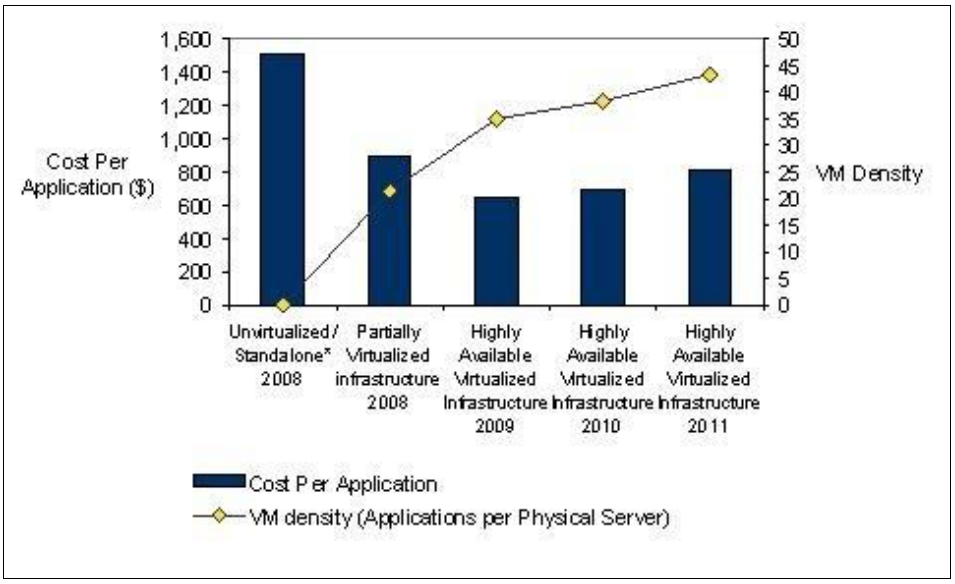


Figura 5. Comparativa del coste de una aplicación instalada en un entorno virtual y no virtual. [6]

Es interesante resaltar cómo se produce una reducción significativa, de hasta el 50% del coste por aplicación, al desplegarse ésta en un entorno virtual. Este factor hace a esta tecnología, si cabe aún, más atractiva para las empresas.

Por ello, este proyecto utiliza la virtualización como herramienta para el desarrollo de un laboratorio *malware* en el que se puedan realizar pruebas aisladas, ágiles y con un mínimo coste, tanto económico como temporal. Ello nos proporcionará datos relevantes a tener en cuenta tanto en el proceso de despliegue de entornos virtuales como en la realización de informes de seguridad.

En definitiva, poder trabajar en un entorno virtual permitirá, tanto a los clientes actuales como a los futuros, realizar una infinidad de pruebas en un medio accesible, ágil, sencillo y escalable. Es un proyecto abierto a cualquier ámbito relacionado con equipos que actualmente son físicos y que pudieran ser virtuales.

[6] http://blogs.vmware.com/virtualreality/2010/01/the-new-economics-of-a-virtualized-datacenter-moving-towards-an-application-based-cost-model.html

1.2. Objetivos

El principal objetivo de este proyecto es el de dotar de un laboratorio virtual al Grupo de Seguridad en las Tecnologías de la Información y las Comunicaciones (COSEC) de la Universidad Carlos III de Madrid para permitir realizar en el mismo cuantas pruebas de *malware* se requieran, y hacerlo de forma ágil y sencilla en un entorno aislado para cada prueba.

En base a este objetivo principal, a continuación se enumeran los principales objetivos secundarios que engloba el proyecto:

1. Instalación y configuración de la infraestructura necesaria para la virtualización de los entornos.
2. Desarrollo de un laboratorio automático de *malware* y de un laboratorio web para la realización de pruebas en entornos virtuales.
3. Instalación y puesta en marcha de los servicios necesarios para pruebas de COSEC.
4. Instalación y configuración de servicios de infraestructura para uso de COSEC.

1.3. Estructura de la memoria

Con el fin de facilitar la lectura y comprensión de la memoria, posibilitando una visión de conjunto, se incluye a continuación un breve resumen de cada capítulo.

Capítulo 1. Dentro de este capítulo se describe el contexto global del proyecto, explicando las líneas generales del mismo y su motivación. Además se presentan los objetivos principales del mismo y un pequeño resumen general de la estructura del documento.

Capítulo 2. A continuación, en el capítulo 2, se pasa a estudiar los diferentes tipos de programas o hipervisores existentes, analizando sus características y diferencias con el fin de explicar la base para la construcción del entorno virtual. El desarrollo será más extenso en el caso de Xen, al ser el hipervisor seleccionado con el fin de proporcionar el entorno deseado. Adicionalmente, se definen las herramientas para la configuración del laboratorio web y los sistemas de detección de *malware* utilizados en el laboratorio automático.

Capítulo 3. El siguiente paso consiste en presentar todos los pasos relacionados con el proceso de diseño de los diferentes entornos, cosa que se efectúa en el Capítulo 3. En el mismo, se presenta la arquitectura global del entorno así como la arquitectura de ambos laboratorios (laboratorio automático y laboratorio web), la definición de los elementos que intervienen en los procesos y los diagramas correspondientes para disponer de una visión global del laboratorio web y del laboratorio automático.

Capítulo 4. En este capítulo se explican en detalle los procesos necesarios que se llevaron a cabo para la implantación final de los laboratorios, desde la instalación de los diferentes elementos que componen los entornos, a las diferentes configuraciones específicas que se implantaron y los detalles de implantación que se han utilizado en cada caso para así conseguir los entornos esperados en cada caso.

Capítulo 5. Como aplicación práctica, en este capítulo se muestran los experimentos *malware* realizados en el entorno del laboratorio automático a modo de evaluación.

Capítulo 6. El Capítulo 6 contiene la descripción de los diferentes procesos relacionados con la gestión del proyecto. Destaca la definición de los requisitos establecidos y un análisis económico y tecnológico del proyecto.

Capítulo 7. Por último, el Capítulo 7 se dedica a exponer las principales conclusiones del proyecto y las posibles líneas futuras que se podrían llevar a cabo.

Glosario. Se muestra el glosario de términos.

Bibliografía. Se especifica la bibliografía utilizada en el desarrollo del proyecto.

ANEXO I. Instalación y Configuración de CloudStack. Explica los pasos a seguir para la instalación y configuración del entorno web con CloudStack.

ANEXO II. Instalación de sistemas operativos Ubuntu. Se describen las acciones necesarias para instalar este tipo de sistemas.

ANEXO III. Creación de plantillas e imágenes en CloudStack y subida de ficheros vhd e iso al entorno web. Se exponen las operaciones que hay que realizar para crear plantillas, subir discos…

Capítulo 2

2.ANÁLISIS

De acuerdo con la enumeración efectuada, en este capítulo se describen los diferentes hipervisores y las diversas herramientas de gestión existentes. Para ello, tras realizar un estudio comparativo de las distintas tecnologías existentes, se efectúa una revisión general de la tecnología de virtualización. A continuación, se expone un breve estudio de la capa de virtualización utilizada para la gestión del hardware, denominado hipervisor, para finalizar con un análisis de la utilización del entorno virtual como entorno de cómputo en la nube o *cloud computing*. Por último, se presenta la herramienta que se utiliza para realizar el análisis de seguridad del laboratorio automático.

2.1. Virtualización

Un concepto que se ha definido en la introducción es el de virtualización, concepto que constituye la base para la realización del proyecto, dado que el mismo se ejecuta, precisamente, en un entorno virtual. Por ello, interesa perfilar de forma más precisa este concepto.

2.1.1. Definición

Se puede definir la virtualización como *"la abstracción de los recursos de una computadora, llamada hypervisor o VMM (Virtual Machine Manager) que crea una capa de abstracción entre el hardware de la máquina física (host) y el sistema operativo de la máquina virtual (virtual machine, guest), dividiéndose el recurso en uno o más entornos de ejecución."* [7]

En cualquier caso, la virtualización constituye una tecnología que ya superó hace tiempo el punto en el que era considerada como una práctica a tener en cuenta, para consolidarse en un *status* en el que se establece como una realidad consolidada, cada vez más demandada por las grandes/medianas empresas a causa de los inmensos

[7] http://es.wikipedia.org/wiki/Virtualizaci%C3%B3n

beneficios que proporciona, frente al sistema tradicional de gestión del hardware y el software. Esta tecnología ha alcanzado su culmen en lo que actualmente se conoce como la "nube" o *Cloud Computing*, con la posibilidad de disponer de la capacidad de poder ejecutar procesos en cualquier lugar del mundo, con plena abstracción del lugar físico desde el cual realmente se estén ejecutando.

Más adelante se profundizará en los conceptos de hipervisor y Cloud Computing.

2.1.2. Tipos de virtualización

La virtualización que se utiliza en el proyecto, y que previamente ya se ha definido, es conocida como *virtualización hardware*. La *virtualización hardware* se hace posible gracias a las tecnologías de virtualización hardware o *Hardware-Assisted Virtualization* como por ejemplo las diseñadas por Intel (Intel-VT) o por AMD (AMD-V). Por lo tanto, este tipo de virtualización está condicionada por la circunstancia de que el hardware esté dotado con la capacidad de separar el entorno hardware del software. Su estructura es muy sencilla: existe un "anfitrión" (la máquina física donde se lleva a cabo la virtualización), un "huésped" (la máquina virtual en sí), y una capa que permite la implantación y puesta en marcha de la máquina virtual, constituida por el hipervisor. Existen diferentes tipos de virtualización en función de dónde y cómo se ejecuten dichos sistemas operativos o huéspedes.

- **Virtualización completa o Full Virtualization**: Este tipo de virtualización permite al huésped realizar una simulación del hardware disponible para representar su entorno de ejecución. En este supuesto, el huésped no sufre ningún tipo de modificación a nivel de núcleo, ya sea porque no se desea o porque no es posible realizar una adaptación del sistema (al no estar diseñado para la arquitectura correspondiente). Cada máquina virtual emulará sus propios dispositivos como si fuesen nativos, lo que conllevará un coste en el rendimiento de la misma. Como aspecto negativo, tendrán un peor rendimiento de I/Os que los sistemas paravirtualizados; pero, por el contrario, no será necesaria la modificación de su núcleo.

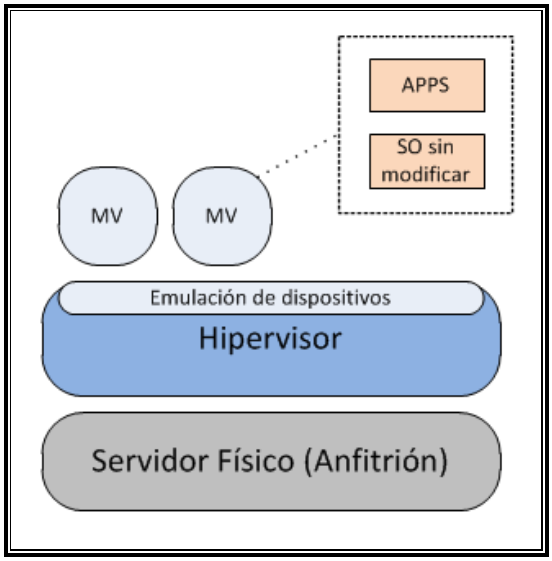


Figura 6. Esquema del funcionamiento de la virtualización completa.

- **Paravirtualización**: En este caso, el huésped sí que tiene que ser modificado a nivel de núcleo, a diferencia del modelo anterior en el que no era necesario. Esta modificación permite al sistema operativo ejecutarse como si no estuviese en un entorno virtual, es decir, como si estuviese ejecutándose de forma aislada al entorno. Este cambio en la máquina virtual o huésped se idea ya que en el caso anterior existen instrucciones que no se pueden comunicar directamente con el HW y tienen que ser gestionadas por el hipervisor. En este caso, la máquina virtual integrará sus controladores con los del hipervisor evitando así grandes cabeceras con destino a los controladores del servidor físico. Compartiendo el acceso físico a los dispositivos se produce una mejora en el rendimiento de las máquinas virtuales. El rendimiento será similar al de una máquina no virtual aunque para ello haya que modificar el huésped para su integración con el hipervisor.

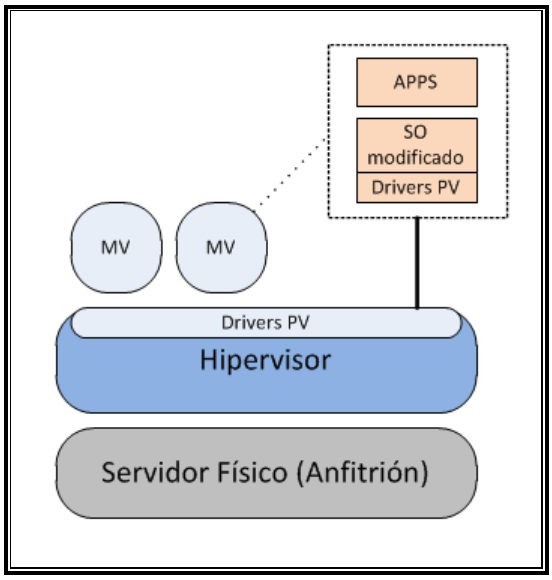


Figura 7. Esquema del funcionamiento de la paravirtualización.

- **Virtualización Parcial.** Existe un tercer tipo de virtualización HW en el que se simula sólo una parte del entorno, frente a lo que sucedía en el caso de la virtualización completa. En este tipo de virtualización es necesario realizar ciertas modificaciones para garantizar el correcto funcionamiento de algunos programas. Este tipo de virtualización no está muy extendida.

2.1.3. Otros tipos de virtualización

Existen otros tipos de virtualización además de la virtualización hardware que se ha analizado anteriormente. A continuación se enumeran los diferentes tipos existentes:

- **Virtualización a nivel de sistema operativo**: Se trata de un tipo de virtualización en el que todos los entornos virtuales *corren* sobre una imagen de un sistema operativo base, retocado para permitir la ejecución de procesos de

varios usuarios. No existe el concepto de hipervisor, sino que esta función la desempeña el sistema operativo base. Los huéspedes se ejecutan en procesos independientes y aislados entre ellos. Es necesario también que el equipo físico disponga de tecnología de virtualización asistida por hardware como por ejemplo Intel-VT o AMD-V.

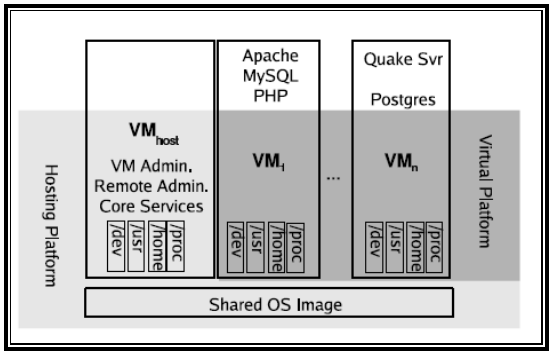


Figura 8. Esquema del funcionamiento de la virtualización a nivel de S.O. [8]

- **Virtualización de Aplicaciones:** En esta variante, la aplicación se ejecuta en el sistema operativo de forma abstracta al mismo. Se trata de un sistema relativamente reciente y cada vez más utilizado debido a sus ventajas.

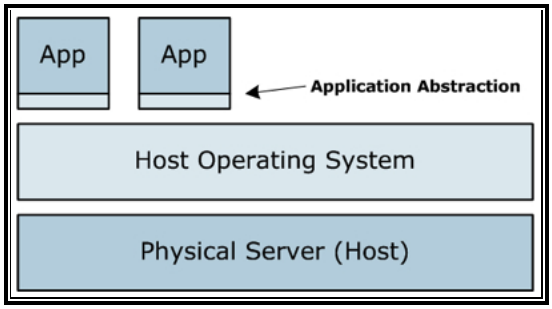


Figura 9. Virtualización de aplicaciones.[9]

- **Virtualización de redes:** Debido a la aparición de la virtualización de sistemas operativos, los hipervisores ofertan redes virtuales para su interconexión sin la necesidad de disponer de una infraestructura de red externa para su uso.

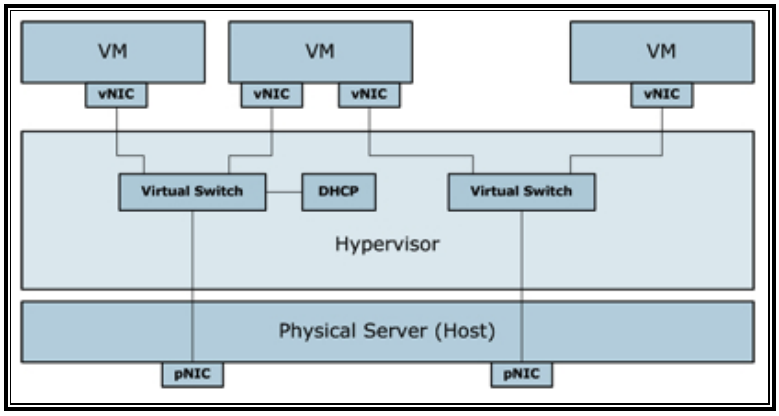


Figura 10. Virtualización de redes.[10]

[8] "Container-based Operating System Virtualization" [1].
[9] http://www.datamation.com/netsys/article.php/3884091/Virtualization.htm

- **Virtualización del almacenamiento:** Bajo esta expresión se conoce al tipo de virtualización en el que se abstrae el almacenamiento real de la forma lógica en la que se presenta al usuario, separando así la forma física de la lógica.

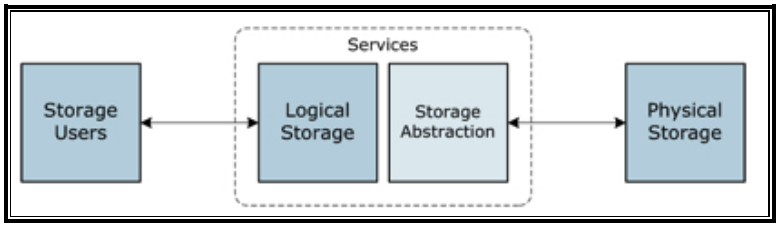


Figura 11. Virtualización del almacenamiento.[10]

A modo de resumen de lo anteriormente expuesto, la *Figura 12. Ratio eficiencia – aislamiento* intenta comparar, para las diferentes tecnologías de virtualización, sus puntos fuertes y débiles mediante la utilización del denominado ratio eficiencia / aislamiento en las diversas tecnologías de virtualización.

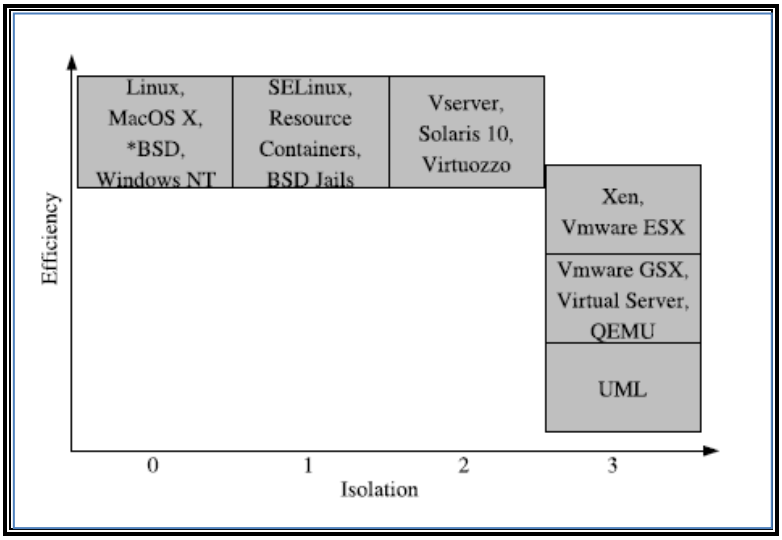


Figura 12. Ratio eficiencia – aislamiento.

También se puede ver en la *Tabla 1. Tabla resumen de técnicas de virtualización* otro resumen de las diferentes tecnologías de virtualización analizadas anteriormente.

	Virtualización completa con transición binaria	Virtualización asistida por hardware	Visrtualización asistida por sistema operativo o Paravirtualización
Técnica utilizada	Transición binaria y ejecución directa de las MVs.	Se realizará una transición al modo root para la ejecución de instrucciones privilegiadas.	Se realizarán *hiperllamadas*.
Modificación del huésped o Compatibilidad	Sin necesidad de modificar el sistema operativo huésped. Excelente	Sin necesidad de modificar el sistema operativo huésped. Excelente	Se realizará una modificación del sistema operativo del huésped para poder

[10] http://www.datamation.com/netsys/article.php/3884091/Virtualization.htm

	Virtualización completa con transición binaria	Virtualización asistida por hardware	Visrtualización asistida por sistema operativo o Paravirtualización
	Compatibilidad.	Compatibilidad.	realizar las *hiperllamadas* por lo que no podrá ejecutarse en otros hipervisores. Pobre compatibilidad.
Rendimiento	Bueno	Correcto	Mejor en ciertos entornos.
Utilizada por:	VMware y Microsoft	VMware, Microsoft y Xen.	VMware y Xen
¿Es el sistema operativo huésped independiente del hipervisor?	Si	Si	Si

Tabla 1. Tabla resumen de técnicas de virtualización[11]

2.1.4. Ventajas de la virtualización

Las principales ventajas de la virtualización, frente al sistema tradicional de gestión de HW y SW se pueden resumir en las siguientes características:

- Separación entre hardware y software.
- Aumento de la escalabilidad del hardware y su optimización.
- Minimizar el espacio en el CPD.
- Ahorro de tiempo en el montaje de máquinas.
- Ahorro de costes (en hardware, en energía…).
- Mejora y sencillez en el mantenimiento de la infraestructura.
- Asignación dinámica de recursos.
- Sencillez en la actualización de parches de los sistemas operativos.
- Administración centralizada y simplificada.
- Tolerancia a fallos.
- Balanceo de carga.
- Homologación de aplicaciones de forma sencilla.

2.2. Hipervisores

Una vez analizada la virtualización y sus posibles escenarios, se profundiza en el término de hipervisor, dado que gracias a sus características, se podrá configurar el entorno del laboratorio *malware*.

2.2.1. Definición

[11] http://virtualization.info/en/news/2007/11/whitepaper-understanding-full.html

Para la creación de los entornos virtuales utilizados a lo largo del proyecto, se utilizará la capacidad de los hipervisores para gestionar el hardware físico de un equipo, capacidad que posteriormente será utilizada por los diversos sistemas operativos virtuales que serán analizados.

Se puede definir un hipervisor, de forma general y simplificada, como una capa intermedia entre el hardware del equipo donde se instala y los diferentes entornos virtuales que se pueden ejecutar sobre él. Es, por lo tanto, un intermediario entre los entornos virtuales o sistemas operativos virtuales y el hardware físico. Las máquinas virtuales, en lugar de usar directamente el hardware como sucede en un equipo físico, interactuarán con el hipervisor que actuará de interlocutor. Los hipervisores también son conocidos como VMM "Virtual Machine Manager" (Gestor de máquinas virtuales).

Una definición más técnica de hipervisor puede ser la que expone Mark Post, Ingeniero de soporte de Novell, en la que define un hipervisor como un sistema operativo que permite la ejecución de múltiples sistemas operativos que se ejecutan sobre el mismo hardware y que no necesariamente tienen que percibir que están compartiendo ese hardware con otros sistemas operativos. [12]

La palabra hipervisor viene heredada de los antiguos supervisores, que era como se conocía a los sistemas operativos originalmente. Cuando nació el supervisor de los supervisores (el supervisor de los sistemas operativos), se le denominó hipervisor.

2.2.2. Tipos de Hipervisores

Existen dos tipos de hipervisores, los denominados de tipo 1 y los denominados de tipo 2. A continuación se analizan ambos tipos de hipervisores.

- Hipervisores de tipo 1:
  Los hipervisores de tipo 1 o nativos, son aquellos en los que el hipervisor se ejecuta directamente sobre el hardware físico del equipo, haciendo uso directo de los dispositivos disponibles, mientras que los sistemas operativos virtuales se ejcutan en una segunda capa superior en las distintas máquinas virtuales. Será necesario, por tanto, que el hipervisor esté en ejecución para que las máquinas virtuales puedan iniciarse y ejecutarse. Los clásicos ejemplos de hipervisor de tipo 1 son XenServer de Citrix, XEN (la versión abierta de XenServer), ESX/ESXi de VMWare, Hyper-V de Microsoft, OVM de Oracle y KVM.[13]

[12] http://www.novell.com/es-es/media/content/linux-for-mainframes-the-hypervisor-defined.html
[13] http://en.wikipedia.org/wiki/Hypervisor

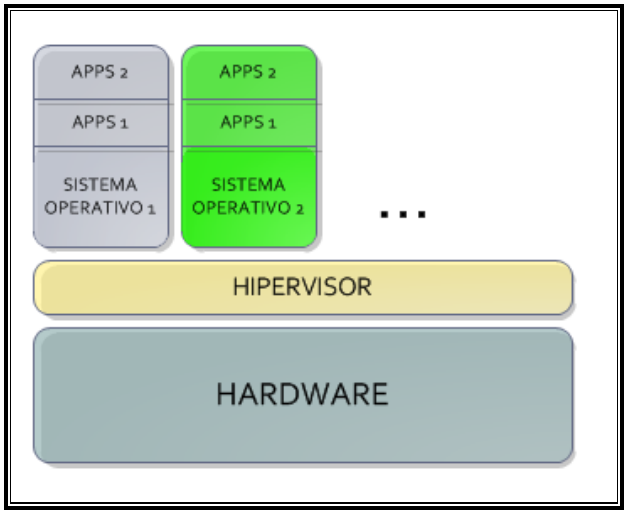


Figura 13. Esquema de un hipervisor de tipo 1.

- Hipervisores de tipo 2:

  Se conoce como hipervisores de tipo 2 o *hosted* (hospedado), a aquéllos en los que el hipervisor se ejecuta sobre un sistema operativo real, no virtual, instalado en el equipo. Este sistema operativo será el *host* o huésped y el hipervisor será un programa más ejecutándose dentro de este sistema operativo. Los sistemas operativos virtuales se ejecutarán en un tercer nivel como procesos independientes dentro del sistema operativo huésped. Unos ejemplos de hipervisor de tipo 2 pueden ser Virtual PC de Microsoft o "VMWare Workstation" entre otros.[13]

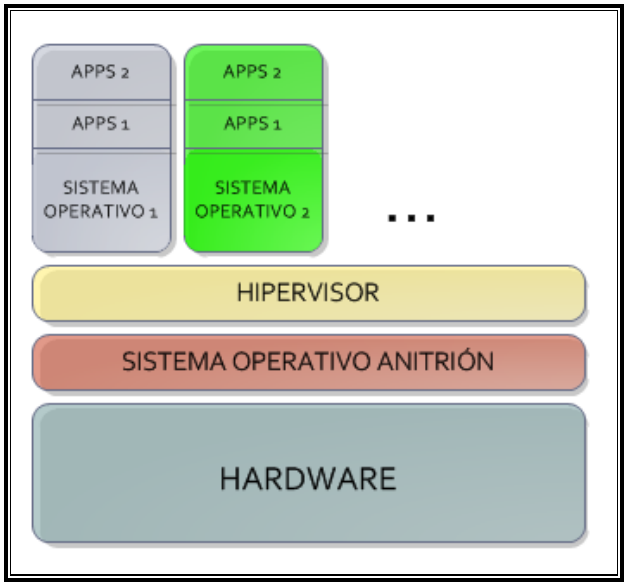


Figura 14. Esquema de un hipervisor de tipo 2.

Nota: KVM (Kernel-based Virtual Machine) ha suscitado muchos debates sobre el tipo de hipervisor en el que se puede incluir. Básicamente se trata de un módulo que se carga en el kernel de Linux que permite al sistema operativo actuar de hipervisor y levantar varias máquinas virtuales. Se ha incluido en la clasificación de hipervisores de tipo 1, ya

que gestiona directamente el hardware sin necesidad de estar en contacto con el sistema operativo base.

2.2.3. Comparativa

Una vez vistos los diferentes tipos de hipervisores, a continuación se realiza un análisis comparativo en relación a los distintos niveles de seguridad y eficiencia de cada uno de ellos.

- Seguridad: A nivel de seguridad, un hipervisor de tipo 1 será, en principio, más seguro[14]. Esto se debe a que, al no producirse instalación de sistema operativo completa, sino que se realiza la instalación de una pequeña (unos pocos MB de tamaño) y segura capa de virtualización, se obtiene una plataforma más fiable y con menores vulnerabilidades. En cambio en los hipervisores de tipo 2, si el sistema operativo base se ve comprometido, todos los sistemas operativos se verán asimismo altamente comprometidos.
  En la *Figura 15. Evolución del número de vulnerabilidades en entornos virtuales* se aprecia la evolución del número de vulnerabilidades en entornos virtuales a los largo de los años:

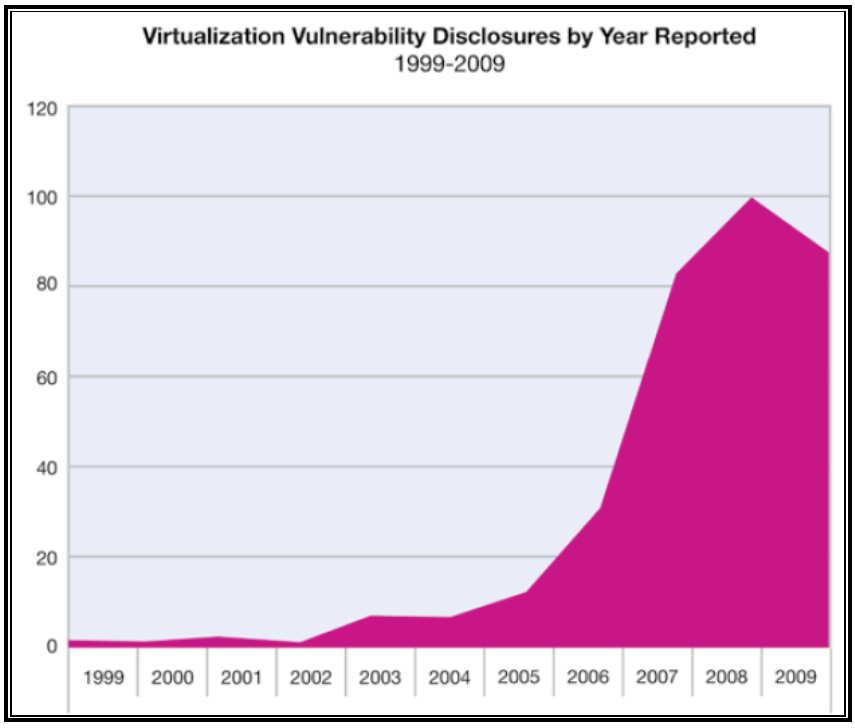


Figura 15. Evolución del número de vulnerabilidades en entornos virtuales[15]

Otro ejemplo interesante es el tipo de vulnerabilidades existentes en entornos virtuales que se muestra en la *Tabla 2. Tipo de vulnerabilidades en entornos virtuales*.[15]:

[14] http://en.wikipedia.org/wiki/Hypervisor#Security_implications
[15] ftp://public.dhe.ibm.com/common/ssi/ecm/en/wgl03003usen/WGL03003USEN.PDF

Tipo	Descripción	Puesto de trabajo %	Servidor %
Anfitrión	Las vulnerabilidades que afectan al sistema operativo del anfitrión en el que el sistema de virtualización está instalado sin la intervención de ninguna de las máquinas virtuales en ejecución.	30.8%	0%
Huésped	Las vulnerabilidades que afectan a una máquina virtual huésped sin afectar al hipervisor o al sistema operativo anfitrión.	26.3%	15.0%
Escape al anfitrión	Las vulnerabilidades que permiten a un atacante "escapar" del entorno de una máquina virtual huésped para afectar al sistema operativo anfitrión en el que el sistema de virtualización se está ejecutando.	24.1%	0%
Aplicación Web	Las vulnerabilidades en aplicaciones Web (normalmente aplicaciones de gestión) que afectan al sistema en el que el navegador del cliente se está ejecutando.	9.8%	10%
Sistema de virtualización	Las vulnerabilidades que afectan al sistema de virtualización en sí, es decir, el entorno virtualizado completo, pero que no surgen de las máquinas virtuales huésped.	4.5%	37.5%
Escape al hipervisor	Las vulnerabilidades que permiten a un atacante "escapar" de una máquina virtual huésped para afectar a otras máquinas virtuales, o al propio hipervisor. En el caso de puestos de trabajo, estas vulnerabilidades no afectan al sistema operativo anfitrión.	3.8%	35%
Consola	Las vulnerabilidades que afectan a las consolas de administración.	0.8%	0%
Servidor Web	Las vulnerabilidades que afectan a un servidor Web que implementa una aplicación web utilizada por el sistema de virtualización.	0%	2.5%

Tabla 2. Tipo de vulnerabilidades en entornos virtuales.[15]

Por último, es interesante observar la *Figura 16. Vulnerabilidades en la virtualización por fabricante*.[15] que compara el número de vulnerabilidades potenciales en los distintos fabricantes de entornos virtuales e hipervisores.

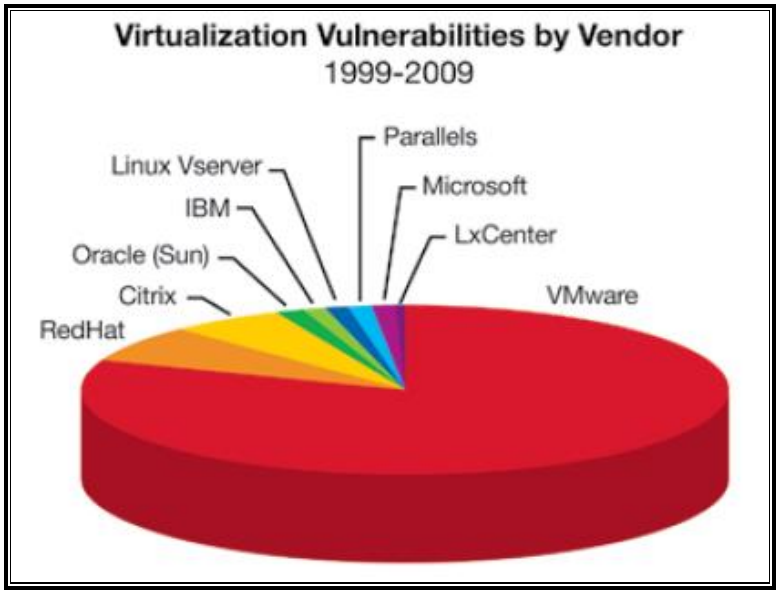


Figura 16. Vulnerabilidades en la virtualización por fabricante.[15]

- Eficiencia: En relación con los aspectos de eficiencia, y al igual que ocurría en el aspecto de seguridad, el hipervisor de tipo 1, ofrece una mejor eficiencia al gestionar directamente los dispositivos físicos, mientas que en el caso de un hipervisor de tipo 2, éste tiene que pasar por el sistema operativo sobre el que se ejecuta para acceder a los dispositivos.

2.3. *Estudio de los hipervisores existentes*

Antes de pasar al estudio de las herramientas de gestión existentes, conviene detenerse en el estudio de los hipervisores. Por ello, en este apartado se aborda un análisis de las diferentes opciones existentes para la construcción del entorno virtual. Por un lado se estudian los distintos hipervisores con los que poder conseguir el entorno virtual, si bien en las próximas secciones, 2.4 y 2.5, se introducirá el término de *Cloud Computing* y los sistemas de gestión virtual para implementar el laboratorio *malware*.

HIPERVISORES		
XEN	Xen	LIBRE
KVM	KVM	LIBRE
VMWARE	vmware	LIBRE

Tabla 3. Hipervisores estudiados.

2.3.1. XEN

Como se ha analizado en apartados anteriores, XEN es un hipervisor de tipo 1, es decir, se trata de una capa que se instala sobre el hardware físico del servidor y que permite el manejo de la siguiente capa superior (virtual), que es donde se ejecutan las máquinas virtuales. Esta separación entre hardware y software la lleva a cabo XEN. Se analizará más a fondo el mundo XEN dado que constituye el hipervisor utilizado para la creación de los entornos virtuales.

- **Historia de Xen.**

La historia de XEN comienza en la Universidad de Cambridge de la mano de Keir Fraser y Ian Pratt como parte de un proyecto de investigación, denominado XenoServer. XenoServer, a su vez, tiene su origen en la palabra latina xenos[16], la cual significa, en el contexto que nos aplica, extraño o desconocido. Este concepto está referenciado a los programas o sistemas operativos que *corren* sobre su hipervisor, los cuales son independientes a él, de alguna manera desconocidos a su entender. Dicho proyecto de investigación consistía en la creación de un sistema para proveer una gestión de recursos ágil y sencilla. Ya se habla en este proyecto de una capa sobre la que se pueden desplegar plataformas de computación: "*how it forms a substrate over which other distributed computing platforms can be deployed*"[17].

La primera versión abierta del producto que conocemos con el nombre de XEN se creó a finales del año 2003. Rápidamente se publicó la versión 2.0 de XEN a finales del año 2004. El año 2006 fue un año donde XEN dio un salto importante: la firma de un contrato de colaboración con Microsoft, en el que se establecen los acuerdos por los que el nuevo hipervisor de Microsoft, Viridian, daría soporte a los huéspedes Linux del hipervisor de XEN y a su vez XEN ofrecería soporte a los ficheros VHD ("Virtual Hard Drive") que se generaban en el hipervisor de Microsoft. Todo ello implicaba que los sistemas operativos Windows comenzarían a ser soportados en XEN.

En 2007 Citrix Systems, Inc. adquiere XenServer por 500 millones de $ y, a principios del año 2010, se presenta la versión 4.0 de XEN. La principal novedad de esta versión consiste en la posibilidad de utilizar como *dom0* un kernel de Linux. La versión del Kernel 2.6.31 fue modificada para dicho propósito.

A principios del año 2011 se presenta la versión 4.1 de XEN, versión que soporta más de 255 procesadores, tiene una mejora importante en estabilidad gracias a unos procesos de regresión automatizados, y mejora el acceso a memoria introduciendo un avance de seguridad en los entornos virtuales entre otros.

[16] http://en.wikipedia.org/wiki/Xenos_(Greek)
[17] Controlling the XenoServer Open Platform, Bibliografía [2]

- **Arquitectura de Xen.**

Xen es la capa que media entre el hardware y los diferentes sistemas operativos. Permite la ejecución de múltiples sistemas operativos simultáneamente ofreciendo un entorno independiente para cada uno de ellos.

A continuación, en la *Figura 17. Arquitectura de Xen.* se muestra un esquema más completo de la arquitectura de Xen donde se pueden apreciar los diferentes componentes y el funcionamiento interno de los mismos.

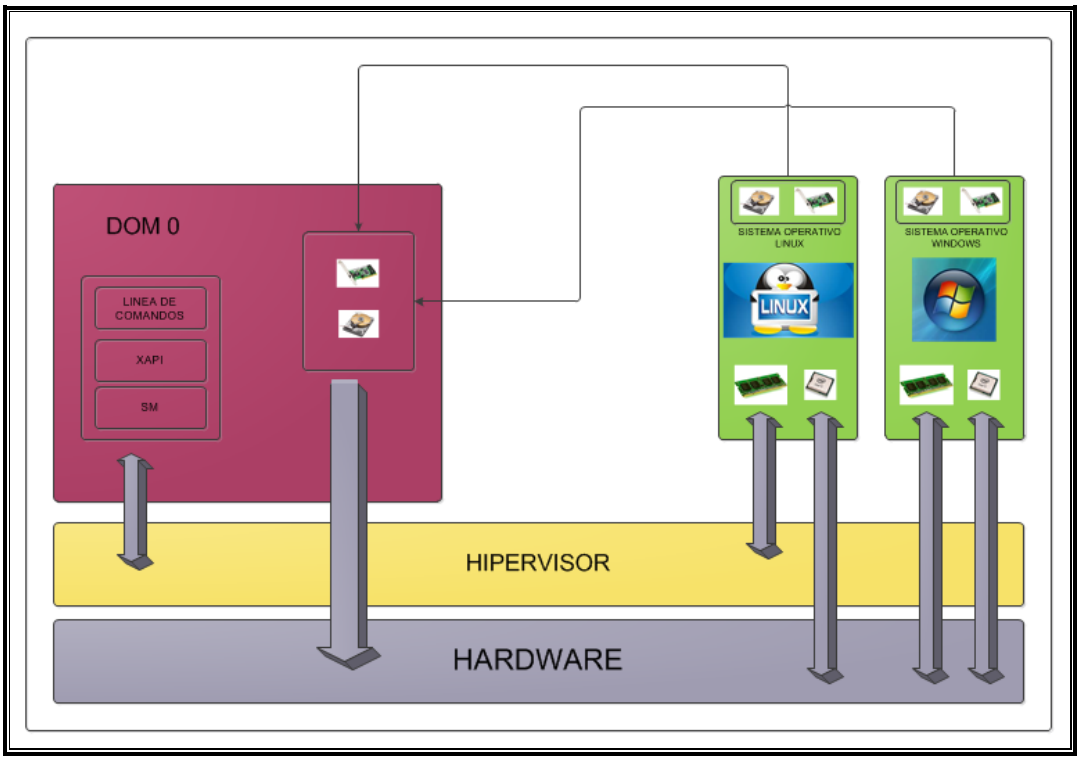


Figura 17. Arquitectura de Xen.

En este esquema, pueden apreciarse cuatro componentes perfectamente diferenciados:

- **Hardware**, es decir, la parte física de toda la solución. En ella se encuentran los diferentes componentes físicos que harán el trabajo final.
- **Hipervisor Xen**, que constituye la capa que separa el entorno físico del virtual. Se ejecuta directamente sobre el hardware actuando de interfaz entre ambos entornos.
- **DOM 0**, o dominio 0. Se trata del primer entorno que se ejecuta en el inicio del sistema. Es una máquina virtual Linux con permisos privilegiados sobre el hipervisor que permitirá el manejo de las diferentes máquinas virtuales para su gestión.
- **DOM U** o resto de dominios, denominación bajo la que se identifican los diferentes sistemas operativos que se van a ejecutar sobre el hipervisor y que no van a disponer de privilegios de acceso al hipervisor. Serán controladas por el Dom0 y son totalmente independientes entre sí. Se podrán levantar sistemas operativos Linux, Windows…

2.3.2. KVM

Por su parte, KVM es una tecnología de virtualización que se puede englobar tanto en los hipervisores de tipo 1 como en los de tipo 2, aunque se podría considerar de tipo 1 al realizar un acceso directo a los dispositivos como parte del sistema operativo en el que reside. Dejando de lado las posibles polémicas, KVM o Kernel-based Virtual Machine es una solución de virtualización para Linux. Se trata de unas extensiones de Linux que se cargan en el *kernel*, lo que genera un hipervisor que permite la creación de una infraestructura de virtualización. La utilización de KVM como hipervisor permite la creación y ejecución de múltiples máquinas virtuales que se presentan como procesos estándar de Linux integrados en el sistema.

A continuación se presenta un esquema básico de la arquitectura de KVM.

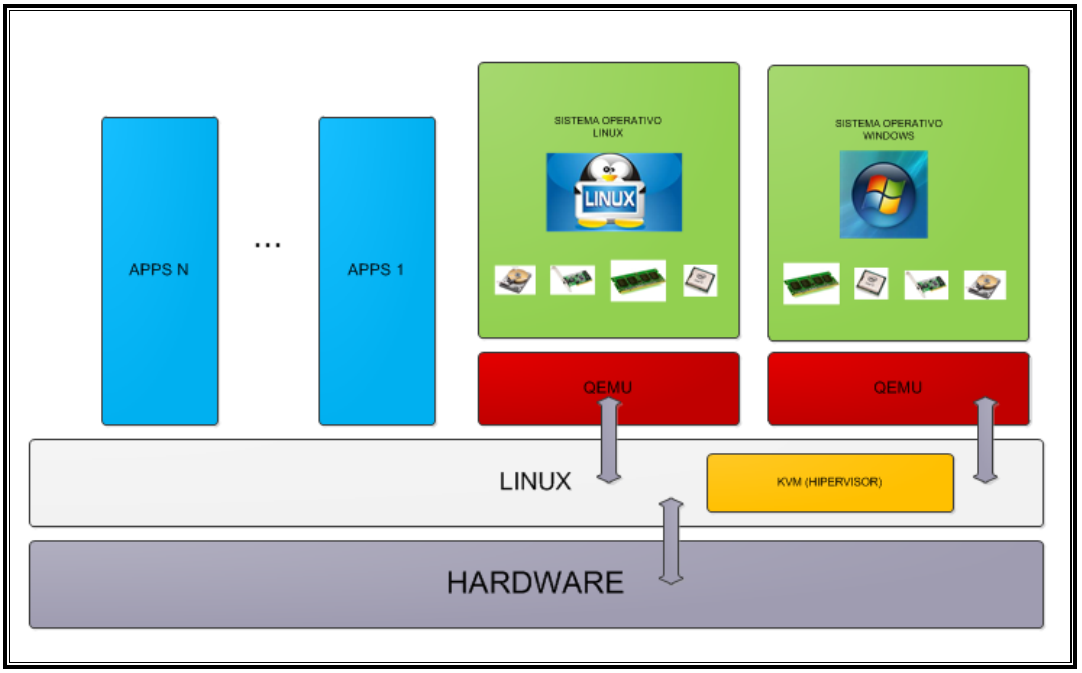


Figura 18. Arquitectura básica de KVM.

Para utilizar KVM como hipervisor, habría que instalar en primer lugar Linux como sistema operativo en el servidor para, posteriormente, cargar el módulo de kvm correspondiente que permite la virtualización del *kernel*. Si se quisiera utilizar las extensiones de Intel o AMD habría que cargar sus correspondientes módulos.

Merece la pena resaltar un nuevo estado que se genera en la máquina Linux para la utilización de máquinas virtuales. Se trata del *guest-mode* o modo invitado, que se añade a los dos modos de operación ya existente *kernel-mode* o modo kernel y *user-mode* o modo usuario.

2.3.3. VMWARE

Al igual que Xen, VMware utiliza una versión de hipervisor de tipo 1. Se instala directamente sobre el hardware sin necesidad de disponer de ningún elemento adicional para controlar y gestionar los dispositivos físicos del servidor. Dispone de una amplia gama de productos, de los que se va a analizar el hipervisor. Los productos de VMware son anteriores a las extensiones de virtualización para el conjunto de instrucciones x86, por los que no requieren de procesadores con extensiones de virtualización habilitados. Pero en caso de disponer de procesador con extensiones de virtualización, el hipervisor está diseñado para hacer uso de las mismas, siendo el rendimiento más eficiente.

La típica arquitectura de un hipervisor VMware es la siguiente:

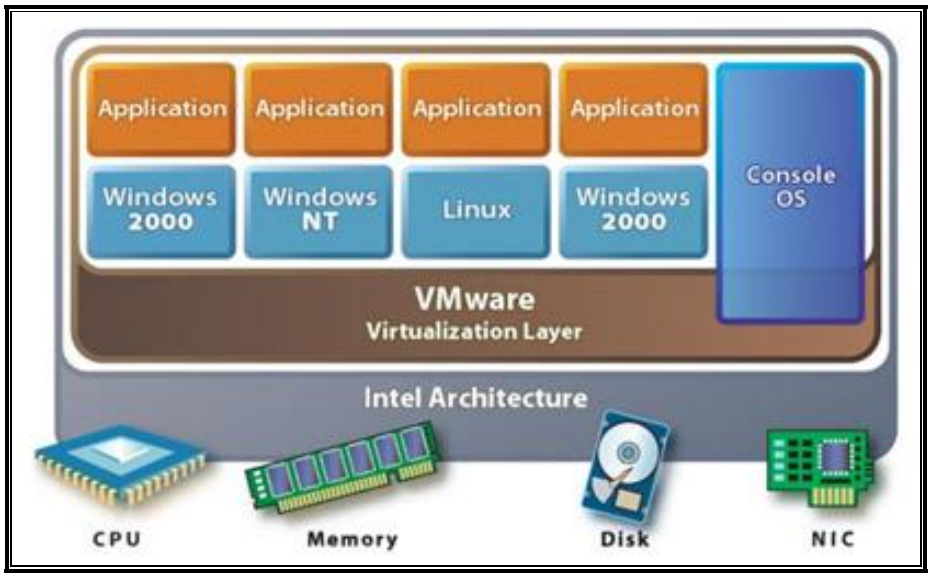


Figura 19. Arquitectura básica de VMware.[18]

VMware dispone de una amplia experiencia en virtualización. Inicialmente se basó en la virtualización de servidores donde es una referencia. En la actualidad se encuentra, al igual que las diferentes empresas de virtualización, ampliando sus objetivos y su mercado a la virtualización de escritorios.

[18] http://www.eginnovations.com/blog/2010/11/07/why-does-your-monitoring-system-need-to-be-virtualization-aware/

2.3.4. Comparativa de los diferentes hipervisores.

Una vez vistos los diferentes hipervisores, se precisa definir las principales diferencias entre ellos.

- Tanto Xen como VMware están basados en un modelo de hipervisor *delgado*, lo que quiere decir que utiliza menos líneas de código. Esto implica que utiliza menos recursos del servidor físico y que dispone de una menor vulnerabilidad ante posibles ataques. En el lado opuesto está KVM que se basa en una distribución de Linux completa, lo que supone una mayor carga en las instrucciones y más código, donde es posible que se produzcan más vulnerabilidades.
- Xen no dispone de drivers de los diferentes dispositivos, mientras que VMware dispone de un almacenamiento de drivers de equipos.
- Xen presenta un buen aislamiento entre las máquinas virtuales a nivel de memoria y CPU, siendo muy bueno a nivel de acceso a disco. KVM presenta también buenos resultados de aislamiento, donde obtiene mejor nota que Xen a nivel de red.
- Xen muestra una buena escalabilidad. Ésta es lineal en función del número de huéspedes, sin verse ninguno de ellos afectado, mientras que en KVM, a medida que van aumentando el número de máquinas virtuales, alguna de ellas se *colapsa*, no presentando una buena escalabilidad.
- Aunque en el desarrollo del proyecto se utilizaron las versiones gratuitas de los productos, las versiones de pago son más económicas en los productos Xen que en los productos VMware.
- Las versiones de KVM y Xen provienen de soluciones de código abierto mientras que las de VMware no.
- VMware ofrece un amplio soporte de sistemas operativos, superior que el de sus competidores aunque los sistemas operativos más importantes (Linux y Windows) están disponibles en todos ellos.

Como resumen final de la situación actual de los diferentes fabricantes de tecnología de virtualización, se puede apreciar, en la *Figura 20. Cuadrado mágico de infraestructura de virtualización Mayo 2010.*, como VMware es un líder mundial pero también cómo Citrix y Microsoft han ido recortando terreno sobre su hegemonía:

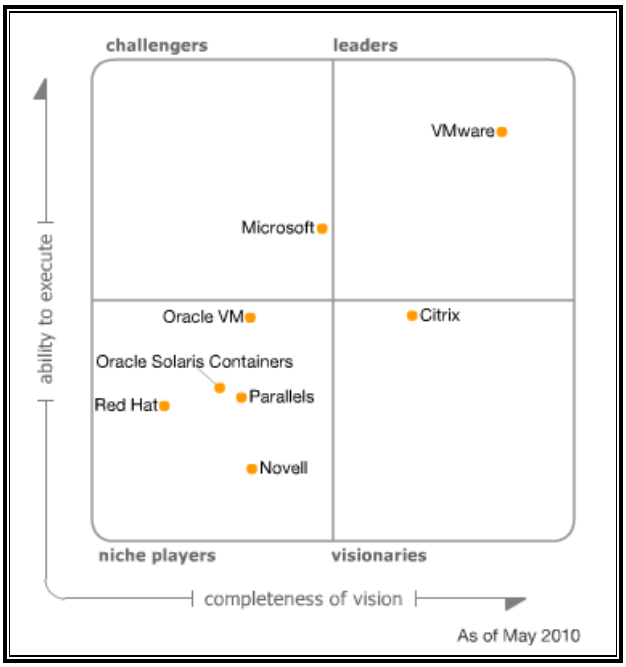


Figura 20. Cuadrado mágico de infraestructura de virtualización Mayo 2010.[19]

Y su comparativa con los datos de 2012:

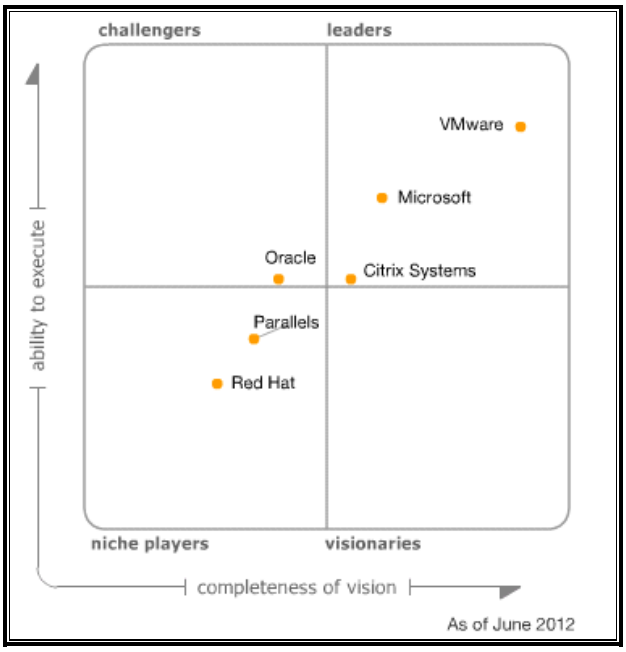


Figura 21. Cuadrado mágico de infraestructura de virtualización Junio 2012.[20]

Se puede concluir que los diferentes hipervisores presentan diferencias en cuanto a diseño y prestaciones. La diferencia entre las diferentes compañías líderes no es muy grande y dependerá del entorno de ejecución para poder diferenciar las diferentes características.

[19] http://www.gartner.com/technology/media-products/reprints/vmware/article4/article4.html
[20] http://www.gartner.com/technology/reprints.do?id=1-1B2IRYF&ct=120626&st=sg

Se utilizará Xen como hipervisor en base a sus características y ventajas y que se trata de un sistema que está en auge y se está popularizando gracias a sus buenas prestaciones y características.

2.4. Cloud Computing

Dentro del estudio, se da un paso adicional, utilizando, para ello los conocimientos adquiridos anteriormente, con el objetivo de analizar la creación de las denominadas *nubes*, que no son sino entornos en los que se llevarán a cabo los diferentes laboratorios de *malware*.

2.4.1. Definición

Se puede definir *cloud computing*, conocida en español como "la nube", como la tecnología que permite ofertar servicios de cómputo y almacenamiento de forma remota, a través de una red.

Esta tecnología tiene la capacidad de ofrecer aplicaciones, servicios y datos bajo demanda sin necesidad de disponer de una infraestructura "propietaria" para poderlo ejecutar. Obviamente, ello supone un ahorro de costes, ya que éste se restringe al propio servicio que se utiliza, sin que alcance a todos los restantes elementos que habría que asumir para poderlo ejecutar en propiedad. Además, no es necesario tener conocimientos de cómo está gestionado o administrado el entorno, ni del lugar en el que se encuentran ubicados los recursos físicos. Existen muchos ejemplos, cada vez más, de empresas que ofertan servicios en "la nube". Los más conocidos son: Amazon EC2, Google, RackSpace, Citrix, VMWare, Microsoft o Verizon. Estas empresas proveen las aplicaciones, infraestructura o escritorios, entre otros, que son accesibles a través de cualquier navegador mientras los diferentes elementos se están ejecutando, procesando o almacenando en los diferentes centros de procesado de datos que poseen los proveedores del servicio.

2.4.2. Tipos de Servicio

Existe una amplia variedad de tipos de servicios que se pueden ofertar en "la nube". Los tres más importantes son:

2.4.2.1. SaaS (Software as a Service).

La peculiaridad de este tipo de nube consiste en la ejecución de aplicaciones en remoto, en la infraestructura de terceros, normalmente a través de Internet con un navegador web. Pueden ser gratuitas o de pago. Algún ejemplo de este tipo podría ser Gmail de Google o Microsoft Office 365.

Las ventajas de utilizar este tipo de servicios son:

- Rapidez para iniciarlas (no hay que instalar o configurar nada).
- Accesibilidad desde cualquier ordenador, en cualquier lugar.
- Aisladas. Si el equipo utilizado se estropea, no se pierden los datos ya que se encuentran almacenados en un entorno aislado y replicado.

Un servicio que surge de éste es el de DaaS (Desktop as a Service) en el que se oferta un sistema operativo completo con aplicaciones a los usuarios.

2.4.2.2. PaaS (Platform as a Service)

Este tipo de *nube* oferta una infraestructura de cómputo con todos los elementos necesarios para ofrecer aplicaciones web sin el coste y la complejidad de su instalación y gestión.

Algunas de las ventajas de este tipo de servicio son:

- Escalable. Los recursos se asignan dinámicamente en función de las necesidades.
- Rapidez. Se pueden desplegar aplicaciones web en cuestión de minutos.
- Sencillez. Se reduce la complejidad en infraestructura y su gestión.

2.4.2.3. IaaS (Infraestructure as a Service)

Este tipo se caracteriza por ofertar todos los componentes necesarios (servidores, red, almacenamiento…) para poder generar una infraestructura completa de cómputo. Se realiza una especie de alquiler de recurso de terceros para instalar y gestionar la infraestructura propia. Algunas de las ventajas serán:

- Escalabilidad. Servicios bajo demanda que se pueden ampliar o reducir dinámicamente.
- Autoservicio de componentes.
- Reducción de costes gracias a la economía de escala y a la agrupación de recursos.
- Pago por uso del servicio.

En este proyecto, se utilizará el modelo IaaS para ofrecer a los distintos grupos un entorno donde diseñar su propia infraestructura de uso. En la *Figura 22. Esquema lógico de cloud computing*. se puede observar de forma global la estructura de los tres servicios estudiados de *cloud computing*.

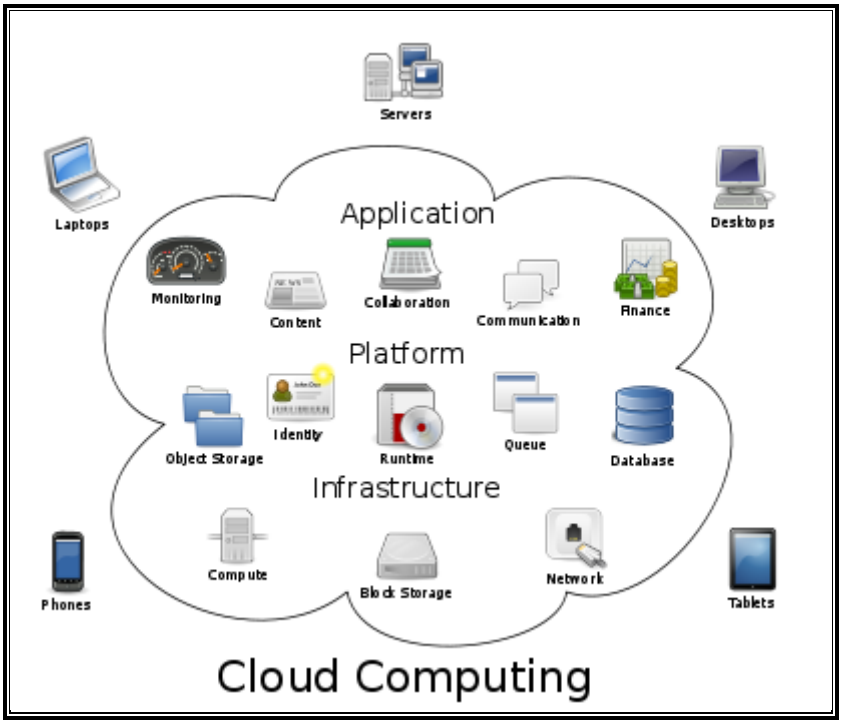


Figura 22. Esquema lógico de cloud computing. [21]

2.4.3. *Tipos de "nubes"*

Existen tres tipos de *nubes* en función del entorno de su utilización. Veamos sus principales características.

2.4.3.1. *Nube Pública*

Las *nubes* públicas son aquellas gestionadas por empresas proveedoras de servicios que ofrecen los diferentes tipos de servicios de *cloud computing* a los usuarios bajo demanda. De esta forma, los usuarios no tienen que comprar ni el hardware ni el software, ni realizar ningún tipo de gestión sobre la infraestructura: simplemente utilizar su tarjeta de crédito para obtener los servicios que deseen. De las *nubes* más conocidas, destacan Amazon o Dropbox.

2.4.3.2. *Nube privada*

Una *nube* privada es una *nube* gestionada por su propietario o grupo de usuarios, ya sea un particular o una empresa. En la misma, se controla tanto la forma en la que se realiza la virtualización de los recursos, como la manera en la que se gestionan. Se suelen utilizar para aprovechar los beneficios de la eficiencia del *cloud computing*, pero controlando los recursos y evitando el uso de éstos por terceros. En este caso serán necesarios conocimientos de virtualización para instalar y configurar los diferentes componentes.

[21] http://en.wikipedia.org/wiki/Cloud_computing

2.4.3.3. Nube híbrida

Una *nube* híbrida, como su propia denominación indica, estará formada por una o más *nubes* públicas y privadas. Se basa en la habilidad de conectar las aplicaciones, recursos, almacenamiento…con una *nube* pública para que interactúen con sus servicios. A continuación se puede observar en la *Figura 23. Tipos de nubes*. una típica *nube* híbrida donde se encontrarán los diferentes elementos.

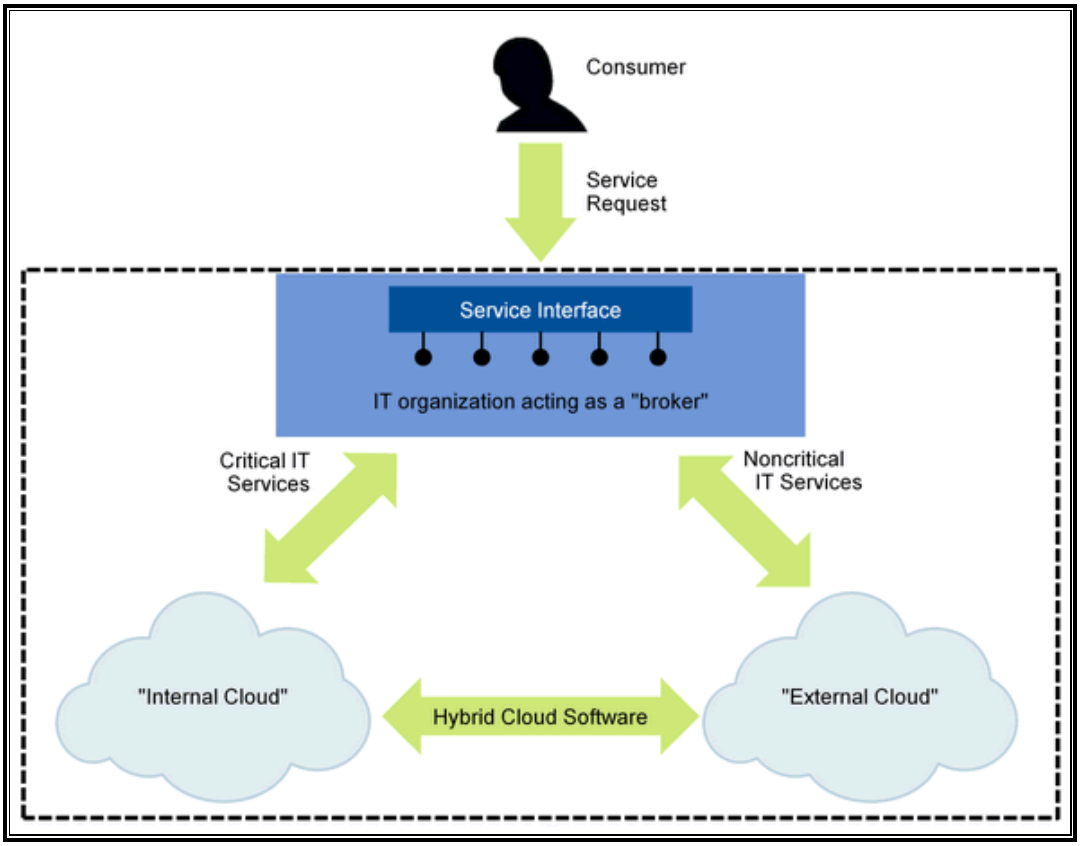


Figura 23. Tipos de nubes. [22]

Son muchas las opiniones sobre los distintos tipos de *nubes*. Pero un aspecto que se resalta en un artículo identifica el concepto de una forma muy curiosa: "***Any Time, Any Place, Any Where, Any Connectivity***"[23]. Este concepto viene a resumir de forma global el concepto de *nube*, en el que uno puede estar conectado en cualquier momento, en cualquier lugar, desde cualquier sitio y con cualquier tipo de dispositivo.

2.4.4. Ventajas y Desventajas del "Cloud Computing"

- Ventajas:
  - Ahorro de costes.
  - Escalabilidad.
  - Recursos casi ilimitados.
  - Ahorro de tiempo en despliegues.

[22] http://www.gartner.com/technology/media-products/reprints/citrix/Citrix_1_19HR5RI.html
[23] http://www.djeek.com/2011/02/hybridcloud/

- Pago por uso.
- Copias de seguridad y recuperación ante desastres.

➢ Desventajas:

- Disponibilidad. No se puede garantizar que esté siempre disponible.
- Protección de datos.
- Privacidad.
- Escasa personalización.
- Sin posibilidad de ejecutar auditorías.

2.5. Estudio de las Herramientas de Gestión existentes

Actualmente existen varias herramientas que gestionan los recursos analizados anteriormente, proporcionando una interfaz *amable* y sencilla para la creación de entornos virtuales. Las herramientas más utilizadas son CloudStack y OpenStack. Interesa, a continuación, analizar ambas herramientas para disponer de un concepto global acerca de las mismas.

Herramientas de Gestión		
ClodStack	cloudstack open source cloud computing	LIBRE
OpenStack	openstack	LIBRE

Tabla 4. Herramientas de Gestión estudiadas.

2.5.1. CLOUDSTACK

CloudStack constituye un software de código abierto que agrupa recursos de cómputo para la construcción de infraestructuras públicas, privadas o híbridas. Se engloba dentro del tipo de *cloud computing* orientado a la infraestructura como servicio, también conocido como IaaS (Infraestructure as a Service). Se trata de una herramienta que permite crear, gestionar y desplegar infraestructuras de cómputo para posteriormente ofertarlas como servicio a los usuarios finales.

CloudStack transforma los recursos hardware disponibles en una *nube* de forma eficaz, manejable y segura. Y todo ello gestionable a través de un interfaz web capaz de realizar todas las operativas correspondientes.

Figura 24. Arquitectura básica de CloudStack.

Los usuarios a los que va dirigida esta solución serían, entre otros, las empresas y los proveedores de servicios, siendo los siguientes sus principales casos de uso:

- Los proveedores de servicios podrían crear sus máquinas virtuales, el almacenamiento correspondiente y las configuraciones de red específicas a través de Internet.
- Las empresas, en su caso, podrían ofrecer una red privada a sus empleados, con las características que se considerasen oportunas para la realización de su trabajo diario en ellas en condiciones más favorables.

CloudStack provee una serie de características que hacen de ésta una herramienta competitiva:

- **No depende del tipo de hipervisor instalado**, es decir, acepta el uso de múltiples hipervisores de forma simultánea e integrada en la misma *nube*. Entre los hipervisores, destacan Xen, KVM, VMWare y XenServer
- **Altamente escalable**, hasta el punto de que se pueden llegar a gestionar decenas de miles de servidores de forma simultánea en centros de procesado de datos distribuidos geográficamente por el mundo a través de una o varias consolas de gestión sincronizadas.
- **Su gestión es automatizada:** CloudStack gestiona automáticamente la red y el almacenamiento de las máquinas virtuales generadas. Para ello, dispone de servidores virtuales que se generan automáticamente para ofrecer los servicios necesarios a las máquinas virtuales.
- **Dispone de una interfaz gráfica de usuario**, así se ofrece una interfaz de gestión gráfica para el administrador del sistema así como otra para los usuarios finales. Ambas interfaces presentarán diferencias en función del perfil correspondiente de cada uno de ellos.
- **Existencia de API's de programación.** Existen APIs de programación de CloudStack para el desarrollo de funcionalidades específicas para cada entorno.

- **Alta Disponibilidad.** Es posible diseñar el entorno para ofrecer una alta disponibilidad del sistema. Tanto a nivel de los servidores de gestión como a nivel de base de datos, de los anfitriones…En definitiva, existen mecanismos para evitar el corte o pérdida de servicio.

2.5.1.1. Arquitectura.

La arquitectura de CloudStack consta de, al menos, dos elementos básicos. Uno de ellos, la máquina donde se ejecutan las instancias de las máquinas virtuales, es decir un hipervisor. El otro, el servidor de gestión a través del que se gestiona todo el entorno. En realidad, con una única máquina podría ser suficiente para generar el entorno, ya que el servidor de gestión puede ser una máquina virtual instalada en el hipervisor donde posteriormente se crearía el entorno final. La estructura mínima necesaria para el desarrollo de CloudStack puede apreciarse en la *Figura 25. Arquitectura básica de CloudStack.*

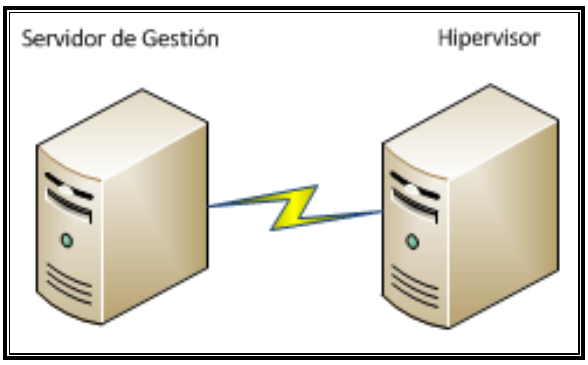


Figura 25. Arquitectura básica de CloudStack.

También podría encontrarse toda la infraestructura embebida en un único servidor como se ha comentado anteriormente.

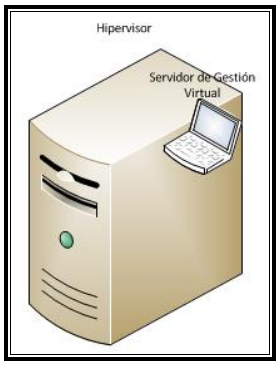


Figura 26. Arquitectura básica CloudStack en un único servidor.

En definitiva, como concepto general, ésta sería la arquitectura mínima y básica que se debería disponer para poder construir una *nube* con CloudStack. Obviamente, la configuración podría llegar a ser mucho más compleja en función de las necesidades y de la infraestructura disponible. Veamos a continuación los dos componentes que forman parte de CloudStack: el servidor de gestión y el hipervisor.

Servidor de Gestión.

El servidor de gestión constituye el principal elemento de CloudStack. Es el encargado de gestionar y administrar los recursos disponibles para ofrecer a los usuarios el entorno deseado.

Sus principales características son:

- Proporciona un interfaz web para los administradores y para los usuarios finales.
- Proporciona los APIs necesarios para poder realizar desarrollos en CloudStack.
- Gestiona la asignación de máquinas virtuales en los diferentes anfitriones o *hosts* disponibles.
- Gestiona la asignación de IPs, tanto públicas como privadas, a las diferentes máquinas.
- Gestiona el almacenamiento de las máquinas virtuales.
- Gestiona las plantillas, imágenes e instantáneas.
- Proporciona un único punto de configuración de la *nube*.

El servidor de gestión es capaz de realizar todas las operativas en función de los datos existentes en una base de datos que él mismo crea y gestiona. En ella, se almacenan todos los objetos de CloudStack, siendo el propio programa el encargado de realizar todos los cambios y/o actualizaciones oportunos. En este proyecto se utiliza como base de datos ***MySQL***. Es posible instalar la base de datos en el propio servidor de gestión o en un servidor independiente. Además, es viable la existencia de varios servidores de gestión *apuntando* a una única base de datos evitando así un posible punto de fallo en caso de que un servidor de gestión fallase o tuviese algún tipo de problema.

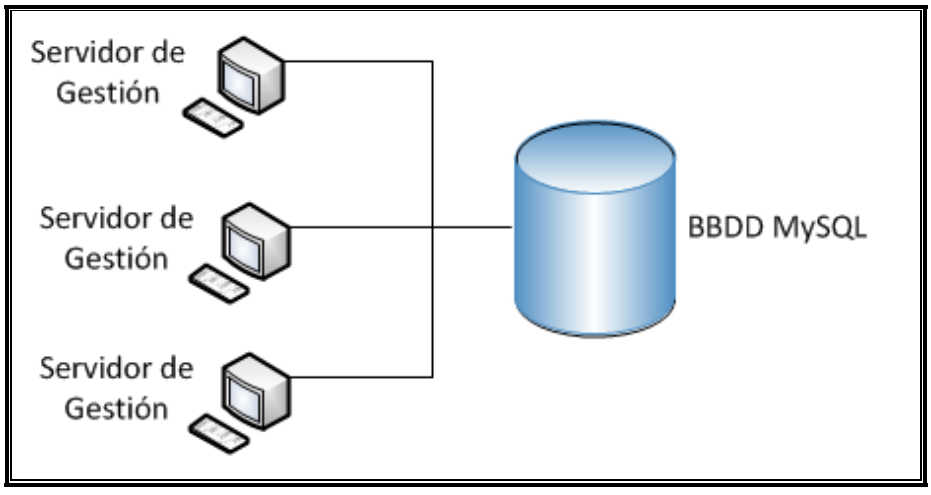


Figura 27. Arquitectura con múltiples servidores de gestión.

Igualmente, es posible disponer de múltiples bases de datos replicadas en el entorno para garantizar una mayor consistencia y una alta disponibilidad del sistema. En este supuesto, la arquitectura de réplica sería muy sencilla, basada en el modelo de maestro / esclavo. Mientras el servidor de gestión ataca a la base de datos primaria, ésta procede a sincronizar la información con la base de datos secundaria que, a su vez, almacenará los

datos en su base de datos local, manteniendo así una base de datos redundante en el sistema. Ambos servidores deberán tener visibilidad entre ellos.

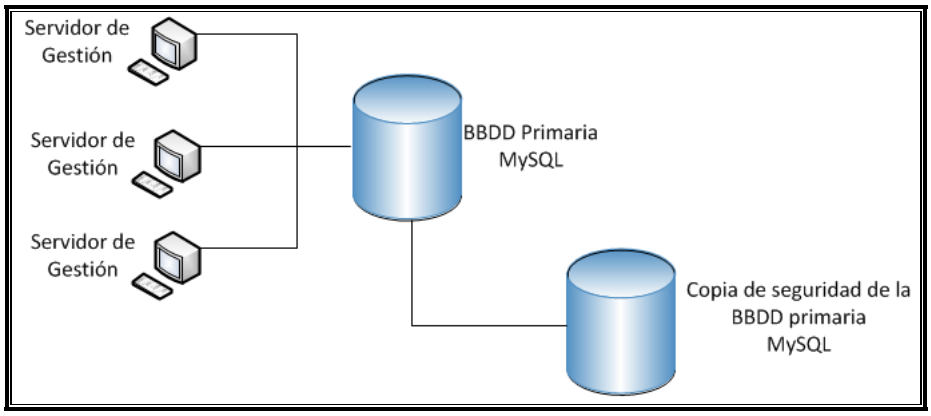


Figura 28. Múltiples servidores de gestión con replicación de BBDD.

Hipervisor.

El hipervisor es el encargado de crear las instancias correspondientes de las máquinas virtuales. En función del hardware disponible, será posible crear más o menos máquinas virtuales. A más recursos hardware, se podrán crear más máquinas virtuales ya que habrá más recursos disponibles (CPU, RAM…) para ello. Es importante que el servidor que actúe como hipervisor tenga unos requisitos hardware mínimos para la creación de una *nube*. Los tipos de hipervisores soportados por CloudStack son:

- Xen
- KVM
- VMWare
- Oracle VM
- Bare Metal

Se pueden apreciar en la *Tabla 5. Principales características ofertadas por CloudStack para los diferentes hipervisores soportados*., las diferentes características soportadas en cada tipo de hipervisor.

Característica	XenServer 6.0.2	vSphere 4.1/5.0	KVM – RHEL 6.2	OVM 2.3	Bare Metal
Control del flujo de red	Si	Si	No	No	N/D
Grupos de seguridad en zonas que utilizan la configuración de red básica	Si	No	Si	No	No
iSCSI	Si	Si	Si	Si	N/D
Uso de fibra como sistema de comunicaciones	Si	Si	Si	No	N/D
Uso de disco local para almacenar los discos	Si	Si	Si	No	Si

Característica	XenServer 6.0.2	vSphere 4.1/5.0	KVM – RHEL 6.2	OVM 2.3	Bare Metal
Alta disponibilidad	Si	Si (Nativa)	Si	Si	N/D
Instantáneas de disco cuando éstos están en el disco local del hipervisor	Si	Si	Si	No	N/D
Uso de disco local como disco de datos	No	No	No	No	N/D
Gestor de equilibrio de carga entre los componentes	No	DRS	No	No	N/D
Migración instantánea manual de una máquina virtual de un anfitrión a otro	Si	Si	Si	Si	N/D
Conservar la gestión de las direcciones IP de tráfico mediante el uso de la red local para comunicarse con el enrutador virtual	Si	No	Si	Si	N/D

Tabla 5. Principales características ofertadas por CloudStack para los diferentes hipervisores soportados.

2.5.1.2. Componentes.

Tras el análisis de la arquitectura básica de CloudStack, se procede a estudiar los diferentes componentes que lo forman, y sus características principales:

- Zonas
- Pods
- Clústeres
- Hosts
- Almacenamiento Primario
- Almacenamiento secundario

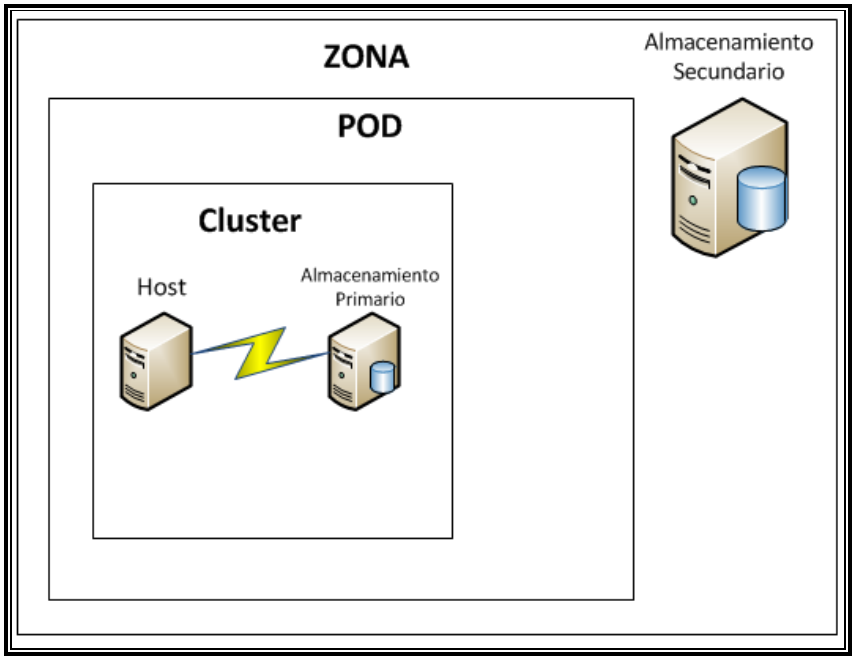


Figura 29. Componentes básicos CloudStack.

Para poder comprender su naturaleza y funcionamiento, se precisa profundizar en el estudio de cada uno de dichos componentes. Sólo de este modo se podrá diseñar la infraestructura correctamente.

Zona.

Una zona es el mayor componente de CloudStack. Se podría identificar una zona con un centro de proceso de datos o, con elementos separados físicamente de la infraestructura. Como se puede ver en la *Figura 29. Componentes básicos CloudStack.*, una zona consta de un sistema de almacenamiento secundario común para todos los elementos de la zona, y uno o varios *pods*, que, a su vez, constan de uno o varios *clústers* o agrupaciones que, asimismo, constan de uno o varios *hosts* y de uno o varios sistemas de almacenamiento primarios. Es decir, engloba a todos los elementos disponibles en un entorno aislado.

Las zonas pueden ser públicas o privadas. La diferencia reside en que las zonas públicas son visibles a todos los usuarios de la *nube* creada, mientras que las zonas privadas son visibles únicamente para aquellos usuarios configurados por el administrador. A su vez, los *hosts* de una misma zona deberán de tener visibilidad directa entre ellos, sin cortafuegos, mientras que los hosts de zonas diferentes únicamente serán visibles a través de la creación de un túnel VPN.

Pod.

El pod constituye el segundo mayor componente de CloudStack. Se define generalmente para identificar un *rack* o una estructura física separada con hosts. Los hosts dentro del mismo pod tendrán la misma sub-red para comunicarse entre ellos. Consta de uno o varios clústeres y uno o varios sistemas de almacenamiento primarios

en función de las necesidades del sistema. Este componente no será visible por los usuarios finales.

Clúster o Agrupación.

En tercer lugar, un clúster es, básicamente, una forma de agrupar diversos hosts con el mismo tipo de hipervisor instalado. Los diferentes equipos físicos que formen un clúster deben de tener las siguientes características:

- Disponer de un HW idéntico.
- Tener el mismo hipervisor instalado.
- Estar en la misma subred.
- Gozar de acceso al mismo almacenamiento primario que será compartido.

Estas características permiten la migración de una máquina virtual de un host a otro host del mismo clúster, sin que se produzca pérdida de datos o deterioro del servicio. Un clúster consta de uno o varios hosts y de uno o varios sistemas de almacenamiento primarios. Pueden coexistir clústeres con hipervisores diferentes en el mismo entorno.

Host.

En cuanto al cuarto componente, un host se corresponde con un servidor. Se trata del componente más pequeño de la infraestructura y será donde se levanten las instancias de las máquinas virtuales que utilizarán los usuarios finales. Proveerá los recursos necesarios a las máquinas virtuales y tendrá las siguientes características:

- Proveer de CPU, RAM, almacenamiento y red a las máquinas virtuales.
- Facilitar la conexión entre ellas y con Internet.
- Pueden residir en la misma zona o en zonas diferentes separadas geográficamente.
- Pueden ser heterogéneos aunque los hosts de un mismo clúster deberán ser idénticos.

La gestión de los recursos de los host la realiza directamente CloudStack. Inspeccionará el HW y detectará dichos recursos para que puedan ser asignados a las máquinas virtuales en función de las necesidades oportunas. Para utilizarlos, bastará con instalar un hipervisor, asignarle una IP fija y darle conectividad contra el servidor de gestión para que éste gestione los recursos disponibles.

Almacenamiento Primario.

En quinto lugar, se definirá el almacenamiento primario, denominación bajo la que se comprende el almacenamiento donde residen los discos locales de las máquinas virtuales y los posibles discos adicionales de datos de los usuarios. Como almacenamiento primario, sería válido cualquier tipo de almacenamiento existente, incluido el almacenamiento local del anfitrión siempre que éste estuviese soportado por la arquitectura del hipervisor. El almacenamiento primario está, a su vez, asociado a un

clúster, pudiendo existir uno o varios por cada clúster. Al tratarse del almacenamiento donde se alojan los discos de las máquinas virtuales, éste deberá tener unas características acordes con el tamaño de la *nube* que se desee crear. Si la *nube* que se quiere crear va a tener un tamaño considerable y no se diseña correctamente el servidor de almacenamiento primario, toda la infraestructura puede verse afectada por el bajo rendimiento del mismo. Es importante mantener un control de la capacidad de este almacenamiento para poder tomar las medidas oportunas antes de que sea demasiado tarde. A continuación se puede observar en una tabla resumen, los tipos de almacenamiento soportados y sus características para los distintos hipervisores:

	VMWare vSphere	Citrix XenServer	KVM	Oracle VM
Formato de los discos, plantillas e instantáneas	VMDK	VHD	QCOW2	RAW
Soporte para iSCSI	VMFS	Clúster LVM	Si, a través de un punto de montaje compartido	Si, a través de OCFS2
Soporte para canal de fibra	VMFS	Si, a través de un repositorio de almacenamiento existente	Si, a través de un punto de montaje compartido	No
Soporte para NFS	Si	Si	Si	Si
Soporte para almacenamiento local	Si	Si	Si	No
Almacenamiento para exceso de provisión	NFS y iSCSI	NFS	NFS	No

Tabla 6. Resumen de características para el almacenamiento primario.

Es posible configurar el almacenamiento primario en modo mantenimiento para realizar las oportunas operativas sobre él, como por ejemplo una copia de éste, un redimensionamiento o un apagado momentáneo del servidor por mantenimiento hardware. Al establecer el almacenamiento en modo mantenimiento, CloudStack realiza las operativas correspondientes para apagar las máquinas virtuales cuyos discos estén en ese almacenamiento. Al recuperarse la operativa habitual, CloudStack devolverá el entorno a su estado anterior.

Almacenamiento Secundario.

En último lugar, el almacenamiento secundario constituye el almacenamiento donde residen el resto de los componentes almacenados que no lo estén en el almacenamiento primario. En consecuencia, el almacenamiento secundario almacenará:

- **Plantillas**. Se trata de plantillas de sistemas operativos diseñados por el administrador a partir de los que se pueden generar máquinas virtuales.
- **Imágenes ISO**. Bajo esta definición se definen discos para instalar un programa cuyo sistema de instalación sea un .iso o imágenes de arranque de sistemas operativos.
- **Instantáneas de volúmenes de disco**, que se podrán utilizar tanto como una copia de seguridad de una máquina virtual, como un disco a partir del que levantar nuevas máquinas virtuales.

Este almacenamiento deberá ser visible para toda la zona a la que pertenezca. Será únicamente accesible vía protocolo NFS[24] y podrán coexistir varios almacenamientos secundarios simultáneamente en la misma zona.

2.5.1.3. Configuración de Red.

La configuración de red se trata de una parte fundamental en la creación de una zona. Las redes se corresponderán con las tarjetas de red físicas de los hipervisores. Cada tarjeta de red puede llevar uno o más tipos de tráfico de red. Existen dos tipos de configuración de red

- Básica
- Avanzada

La configuración de red básica configurará una red predeterminada donde no será posible la configuración de ningún parámetro. Únicamente se utilizará para entornos básicos donde no sea necesaria ninguna configuración concreta. Sin embargo, la configuración avanzada de red proporcionará una mayor flexibilidad a la hora de crear y gestionar las zonas.

A continuación se describen los 4 tipos de tráfico de red configurables:

- **Tráfico de usuario**. Es el tráfico de red generado por el uso de las máquinas virtuales para comunicarse entre ellas. Este tráfico puede ser compartido o aislado. En caso de ser compartido, todas las máquinas virtuales tendrían contacto entre ellas pudiéndose aislar con, por ejemplo, grupos de seguridad. En el caso de una red aislada, se generarán VLANs[25] para la separación de los diferentes tráficos de usuario.
- **Tráfico de Gestión.** Se denomina así al tráfico que se genera entre los diferentes componentes de CloudStack, como la comunicación entre los diferentes hosts o la comunicación de cualquier elemento con el servidor de gestión.

[24] NFS o Network File System es un protocolo de ficheros de sistema distribuidos que permite el acceso a ficheros a través de la red de una forma similar a como se acceden a ficheros en un almacenamiento local. http://en.wikipedia.org/wiki/Network_File_System

[25] VLAN o Virtual Local Area Network es una manera de partir una red física para crear subredes aisladas entre sí.

- **Tráfico público.** Se trata del tráfico de red que se genera cuando las máquinas virtuales se conectan con Internet. Será necesaria la reserva de IPs públicas para que las máquinas virtuales de las diferentes redes puedan acceder a Internet. Se implementará el protocolo NAT[26] para que se pueda llevar a cabo.
- **Tráfico de Almacenamiento.** Este cuarto tipo de tráfico es el que se genera entre los servicios de almacenamiento primario y secundario con los anfitriones o *hosts*.

Como se ha comentado anteriormente, este tipo de tráfico puede ir por una interfaz de red física separada del resto o pueden convivir en una misma.

2.5.2. OPENSTACK

Al igual que CloudStack, OpenStack es un software de código abierto que agrupa recursos de cómputo para la construcción de infraestructuras públicas, privadas o híbridas. Está orientado principalmente a empresas y a proveedores de servicio que deseen desplegar entornos a gran escala de *cloud computing*.

Las características de OpenStack son las siguientes:

- Gestión del ciclo de vida de las máquinas virtuales.
- Gestión de los recursos de cómputo de la *nube*.
- APIs de desarrollo.
- Independencia del hipervisor utilizado. Ofrece soporte para Xen, KVM, UML y VMware.

Al igual que en el caso anterior, se dispone de una interfaz web desde la que se podrán gestionar los diferentes recursos del entorno. A continuación se puede observar un esquema básico del funcionamiento de Openstack.

[26] NAT es un protocolo por el que un enrutador puede permitir la salida de una serie de datos fuera de una red privada. http://en.wikipedia.org/wiki/NAT_Port_Mapping_Protocol

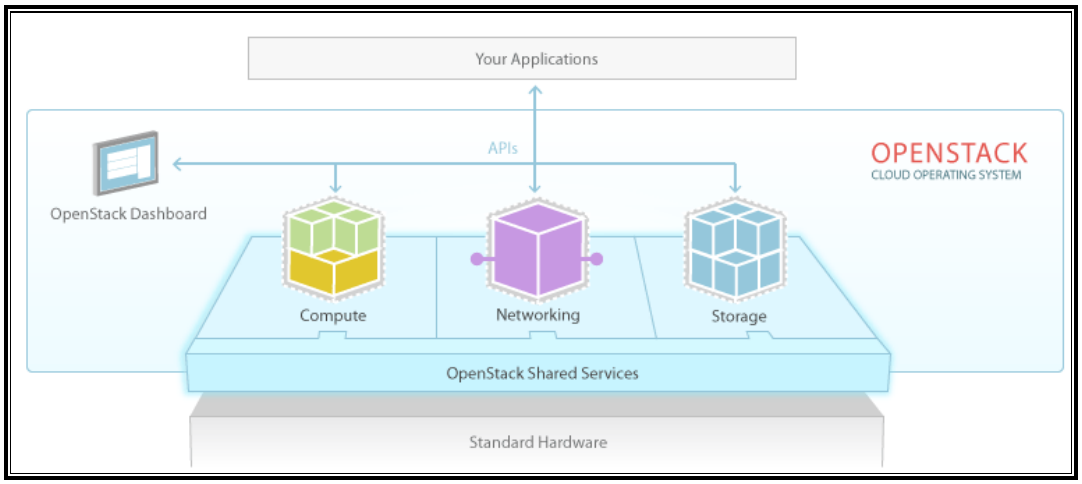


Figura 30. Esquema básico de OpenStack.[27]

2.5.2.1. *Arquitectura*

La arquitectura básica de OpenStack se puede apreciar en la *Figura 31. Arquitectura básica de OpenStack.*:

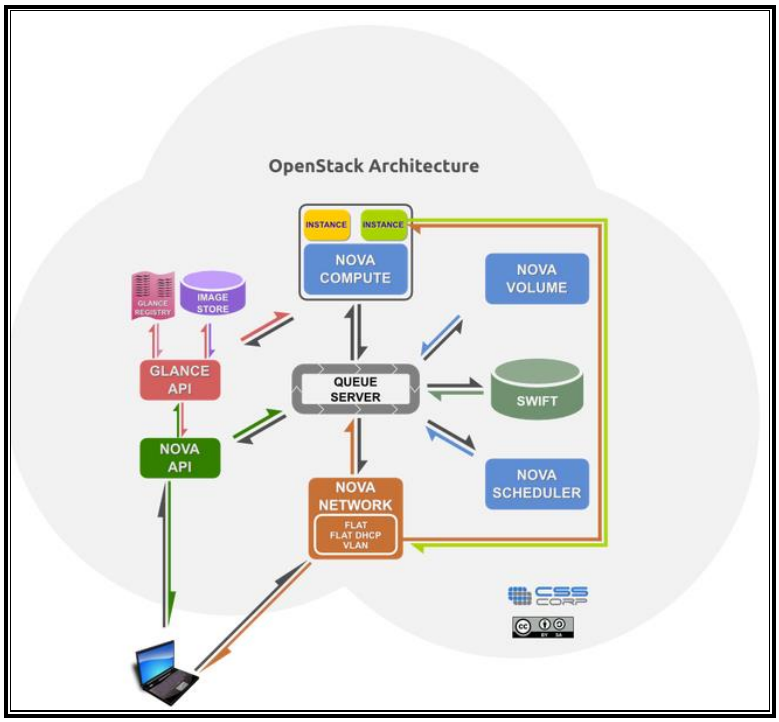


Figura 31. Arquitectura básica de OpenStack.

En ella se pueden observar los diferentes componentes que forman la estructura de OpenStack. Al igual que en el caso anterior, esta solución puede instalarse en un único equipo, adicional al hipervisor, o puede disgregarse en diferentes componentes separados en diferentes máquinas. En el siguiente apartado se analizarán los diferentes componentes de OpenStack.

[27] http://www.openstack.org/software/

2.5.2.2. Componentes.

Los principales componentes de OpenStack son los siguientes:

- Nova: Es el componente que actúa como el cerebro de la infraestructura. Como tal, gestiona los recursos de cómputo del entorno, las redes, las máquinas virtuales…Todas las transacciones deberán pasar por este elemento para su validación y configuración.
- Swift: Este componente se encarga de gestionar el almacenamiento de la infraestructura. Se trata de un sistema de almacenamiento distribuido para datos estáticos, como los de las máquinas virtuales. La distribución de los datos se puede llegar a realizar en múltiples discos dispersos por la infraestructura, soportando múltiples plataformas de almacenamiento. También sería el componente encargado de gestionar las copias de imágenes de sistemas operativos o *snapshots*.
- Glance: Con OpenStack Glance, es posible gestionar el servicio de imágenes. Tiene la capacidad de almacenar un sistema operativo o un *snapshot* del mismo como imagen para el despliegue de nuevas máquinas virtuales. Estas plantillas van a poder ser generadas tanto por el administrador como por los propios usuarios.
- Horizon: Ofrece la implementación de un interfaz web de usuario que controla los diferentes servicios de OpenStack. Constituye el punto de acceso de los administradores y usuarios para gestionar y controlar sus recursos en la *nube.* Tiene un portal de presentación diferente para los administradores ya que en él tendrán una visión general de toda la infraestructura y recursos, mientras que la de los usuarios únicamente presenta los recursos que les hayan sido asignados.

2.5.3. Comparativa OpenStack vs CloudStack

Actualmente, no cabe duda de que ambas constituyen dos de las grandes plataformas de código abierto orientadas a *cloud computing.* Veamos a continuación algunas de las diferencias que enfrentan o caracterizan a ambas versiones.

- En cuanto a su origen, OpenStack surgió originalmente de un proyecto de la NASA y Rackspace, respaldado también por grandes compañías como IBM, HP o Dell. Estos proyectos no forman parte de OpenStack pero lo alimentan. Por otro lado, CloudStack se rige por los estándares del Apache Software Foundation, con el apoyo de Citrix y otras 50 empresas de tecnología.
- CloudStack posee un sistema de instalación y una documentación mucho más sencilla y completa que OpenStack. Mientras que CloudStack ofrece un único punto de instalación, OpenStack dispone de diferentes repositorios con diferentes versiones que se van actualizando constantemente. Este factor, aun siendo interesante, evita una consolidación estable de la infraestructura.
- OpenStack ofrece todas las características para los hipervisores basado en Xen y KVM, mientras que las funcionalidades son limitadas para Hiper-V, XenServer y VMware. CloudStack por su parte, ofrece prácticamente las mismas funcionalidades para KVM, Xen, VMware y Oracle VM, siendo este último un elemento diferenciador con respecto a su competidor. Además presenta soporte para servidores *bare-metal*.

- Tanto OpenStack como CloudStack ofrecen una solución escalable y centralizada.
- Openstack ofrece una amplia variedad de formatos de discos (vhd, vdi, vmdk y ovf), mientras que CloudStack únicamente ofrece soporte de discos en formato vmdk, vhd y qcow2.
- A nivel de red, CloudStack ofrece una mayor variedad de sistemas como OpenFlow, VLAN o *flat network*.
- CloudStack presenta una serie de asistentes que resultan muy sencillos y útiles para los usuarios. Además presenta un sistema de monitorización de recursos que OpenStack no dispone.
- CloudStack ofrece migración en vivo de máquinas virtuales entre hosts a través del interfaz web, característica que no dispone OpenStack.

2.6. Sistemas SIEM (OSSIM)

OSSIM (*Open Source Security Information Management*) es una herramienta de código abierto de lo que se conoce como ***Security Information and Event Management*** (SIEM). Las soluciones SIEM nacen, principalmente, de la combinación de dos productos: SIM (*Security Information Management*) y SEM (*Security Event Manager*). Estas herramientas proveen un sistema cuyo objetivo es el de ayudar a los administradores de redes a mantener la seguridad en los ordenadores, a detectar las intrusiones en los mismos y a prevenirlas.[28]

Este conjunto de herramientas, permiten obtener una visión general del estado, a nivel de seguridad, en el que se encuentra un sistema. Además, en base a la correlación de la información almacenada, hace de ella una potente herramienta de prevención de intrusiones en los sistemas.

2.6.1. Evolución de los sistemas SIEM

Antes de convertirse los sistemas SIEM en las potentes herramientas de monitorización, detección y prevención que son en la actualidad, éstas evolucionaron de diferentes tecnologías que se enumeran a continuación:

- LMS o *Log Management System*: Sistema basado en la recolección de ficheros de *log* o de bitácora de sistemas operativos, aplicaciones…en un punto central donde poderse consultar y analizar.
- SLM/SEM o *Security Log/Event Management*: Se trata de un sistema similar al LMS pero orientado a registros de seguridad. Detectarán y notificarán aquellos eventos que se consideren importantes en relación a la seguridad.

[28] http://en.wikipedia.org/wiki/OSSIM

- SIM o *Security Information Management*: Se trata de un sistema de gestión de activos pero incorporando información sobre seguridad.
- SEC o *Security Event Correlation*: Este sistema busca patrones en los ficheros de bitácora con el objetivo de lanzar alertas cuando se produzca una secuencia concreta de eventos.
- SIEM: Los sistemas SIEM engloban en un único producto las anteriores tecnologías, convirtiéndose en un sistema de gestión de información de seguridad.

En definitiva, se trata de un sistema que *observa* lo que está sucediendo en nuestra red a través de cualquier fuente de información o de control de seguridad.[29]

2.6.2. Características de los sistemas SIEM

A continuación se presentan las principales características de este tipo de sistemas:

- **Recopilación de datos**: Se realiza una recopilación de datos del mayor número posibles de fuentes como puede ser de la red, de datos de seguridad, de bases de datos, de aplicaciones…para realizar una consolidación de los mismos evitando la pérdida de datos.
- **Correlación**: Provee la capacidad de utilizar diferentes técnicas de correlación integrando las diferentes fuentes de datos para transformar los datos en información de utilidad.
- **Alertas**: Generación de alertas en base a la recolección de los eventos y a su correlación. Éstas se pueden enviar a un panel de control o, mediante otras vías como el correo electrónico, a los administradores.
- **Panel de Control**: Transforma los datos almacenados en gráficos informativos donde poder identificar patrones o actividad poco común.
- **Cumplimiento**: Los sistemas SIEM se pueden utilizar para recopilar datos para realizar informes de seguridad, auditoría…
- **Retención**: Utiliza el almacenamiento histórico de los datos para facilitar la correlación de los mismos y para cumplir con la normativa de retención de datos correspondiente.

2.6.3. Sistemas HIDS

Los sistemas SIEM pueden tener integrados varios sensores para detectar y prevenir posibles intrusiones. Dentro de los sensores que se pueden integrar, en este proyecto se ha optado por configurar un sensor denominado HIDS o **Host-Based Intrusion Detection System**. Los sistemas HIDS monitorizan y analizan las *tripas* de un sistema, enviando la información correspondiente al servidor SIEM.

[29] SIEM for beginners, Alienvault ™ https://alienvault.bloomfire.com/posts/556521-a-beginner-s-guide-to-siem/public

2.6.3.1. Funcionamiento de los sistemas HIDS

Su funcionamiento se basa en la premisa de que un atacante que realice una intrusión en un sistema, deja, en un porcentaje elevado de los casos, un rastro. Los sistemas HIDS son capaces de monitorizar todos los ficheros, claves de registro, eventos… de un sistema analizando los posibles cambios que se produzcan en ellos. Si se detecta un cambio en alguna de las estructuras analizadas y configuradas a tal efecto, se enviará una alerta al servidor SIEM para notificar a los administradores de que se está produciendo una anomalía en el sistema.

Una vez instalado e iniciado, los sistemas HIDS realizan un escaneo de los objetos configurados, almacenando la información en una base de datos local. Esta base de datos tarda un tiempo en configurarse ya que ésta consta de un sistema de encriptación de cada objeto monitorizado con la finalidad de que ningún atacante pueda realizar cambios en la base de datos. Es posible monitorizar el *checksum* de los ficheros, su tamaño, fecha de modificación…limitando así las acciones de los atacantes.

Una vez la base de datos está construida y es segura, el sistema HIDS escaneará el entrono cada cierto tiempo, configurable por los administradores, y generará las alertas correspondientes en caso de que un fichero o cualquier aspecto sospechoso se detecte.

2.6.3.2. Ventajas y Desventajas de los sistemas HIDS.

Las ventajas fundamentales de estos sistemas son las siguientes:

- Se trata de herramientas precisas. Escanean los parámetros configurados en el sistema para detectar cualquier modificación.
- Aunque los datos fuesen cifrados, los daos son analizados en el cliente en formato texto plano, por lo que es transparente.

Las principales desventajas son:

- Para aumentar la precisión y seguridad en los sistemas, es necesario desplegar un agente en cada sistema.
- Si el atacante es capaz de reconfigurar el agente, el sistema quedará comprometido.
- El agente consume ciclos de CPU, configurable, pudiendo afectar al rendimiento del sistema.

2.6.4. OSSEC

Como sistema HIDS se ha optado por seleccionar el agente de OSSEC para sistemas Windows. El agente OSSEC es un programa de código abierto que se instala en las máquinas a monitorizar con el objetivo de analizar el sistema operativo en busca de amenazas una vez ejecutado el *malware* en el entorno aislado.

Nace en 2004 de la mano de Daniel B. Cid para, tras años de evolución, ser adquirida en 2009 por Trend Micro, una empresa especializada en seguridad.

A continuación se muestran las principales características de este sistema:

- **Requisitos de cumplimiento**: Ayuda a los clientes a satisfacer el cumplimiento específico de auditorías, normativa…Con OSSEC, es posible detectar y alertar sobre las modificaciones no autorizadas del sistema de archivos y el comportamiento malicioso basado en las entradas de los archivos de registro de productos.
- **Soporte multi-plataforma**: Es posible realizar la monitorización de diferentes plataformas como Linux, Windows, Mac…
- **Alertas en tiempo real configurables**: Es posible configurar el agente para que los incidentes en objetos considerados prioritarios se realice en tiempo real. Dispone de integración con smtp, sms… para el envío de alertas.
- **Integración con la infraestructura actual**: OSSEC se integra perfectamente en los sistemas SIM/SEM/SIEM existentes.
- **Gestión centralizada**: Adicionalmente, existe un servidor de gestión centralizada donde aplicar políticas o definir estrategias de gestión.
- **Monitorización mediante agente o sin él**: Permite la monitorización mediante la instalación de un agente en el sistema o no. Sistemas como enrutadores o cortafuegos podrán ser monitorizados con OSSEC sin necesidad de instalar un agente en ellos.

Todo esto hace de OSSEC una potente herramienta que será de gran utilidad en la detección de intrusiones en los sistemas Windows que inicialmente se desplegarán en el laboratorio, pudiendo en un futuro incluir sistemas operativos Linux de forma simultánea sin necesidad de realizar ninguna modificación.

2.6.4.1. Arquitectura OSSEC

La arquitectura que se utiliza en el laboratorio es muy sencilla. Inicialmente se incluyen únicamente sistemas Windows con agente. No se utiliza la opción sin agente. A continuación se muestra la arquitectura utilizada, donde se muestran las máquinas Windows con el agente instalado y un servidor SIEM central donde se recogen las alertas de las máquinas desplegadas:

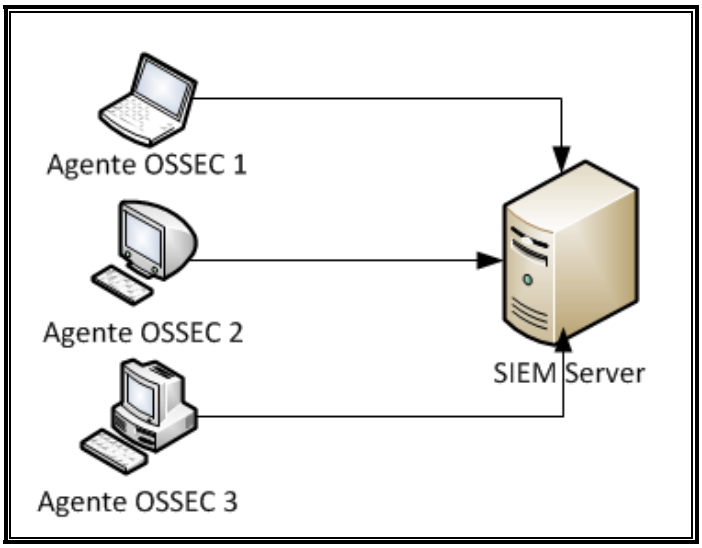


Figura 32. Arquitectura básica de OSSEC.

2.7. Requisitos de usuario

En este punto se procede a estudiar los distintos tipos de requisitos de usuario que se han establecido en la definición del proyecto para, desde los mismos, poder inferir los diferentes requisitos software. Se diseña una tabla por cada requisito de usuario, diferenciando, a su vez, el requisito software funcional y no funcional. También se especifican los requisitos de usuario, mostrando un código de identificación numérico y una breve descripción del mismo.

Requisito de Usuario	RU-01
Descripción	El sistema permite la creación de un laboratorio automático que posibilitará el estudio del *malware* de forma ágil y flexible.

Tabla 7. Requisito de usuario 01

Requisito de Usuario	RU-02
Descripción	El sistema permitirá la creación y gestión de distintos nodos con diferentes sistemas operativos para su estudio.

Tabla 8. Requisito de usuario 02

Requisito de Usuario	RU-03
Descripción	El usuario podrá definir los recursos software específicos que configuran cada uno de los nodos del laboratorio y sus parámetros en función de las necesidades de cada situación.

Tabla 9. Requisito de usuario 03.

Requisito de Usuario	RU-04
Descripción	El usuario podrá definir los recursos hardware específicos que configuran cada uno de los nodos del laboratorio y sus parámetros en función de las necesidades de cada situación.

Tabla 10. Requisito de usuario 04

Requisito de Usuario	RU-05
Descripción	El sistema tiene la capacidad de crear diferentes entornos de red en función de las necesidades. Dichas necesidades pueden ser: la creación de entornos aislados, de entornos conectados al exterior, o de entornos mixtos.

Tabla 11. Requisito de usuario 05.

Requisito de Usuario	RU-06
Descripción	Cada uno de los entornos creados deberá atender a las necesidades de seguridad oportunas. El sistema garantizará el aislamiento entre distintos entornos.

Tabla 12. Requisito de usuario 06.

Requisito de Usuario	RU-07
Descripción	El sistema debe ser tolerante a fallos.

Tabla 13. Requisito de usuario 07.

Requisito de Usuario	RU-08
Descripción	El usuario podrá gestionar los distintos entornos de forma sencilla e intuitiva.

Tabla 14. Requisito de usuario 08.

2.8. Requisitos software

Los requisitos de usuario anteriormente expuestos implican, a su vez, la definición de otros correspondientes requisitos de *software*, que son los que se analizan en la presente sección. Dichos requisitos *software* se especifican mostrando un código de identificación numérico y una breve descripción del mismo.

Requisito Software Funcional	RSF-01
Descripción	El estudio del comportamiento del *malware* requiere la creación de un entorno virtual que permita la ejecución simultánea de distintos entornos.

Tabla 15. Requisito software funcional 01.

Requisito Software Funcional	RSF-02
Descripción	La creación de los distintos nodos necesita de un repositorio de imágenes ISO preparado con configuraciones por defecto de los sistemas operativos (plantillas) deseados.

Tabla 16. Requisito software funcional 02.

Requisito Software Funcional	RSF-03
Descripción	El sistema debe permitir subir, borrar y descargar las diferentes imágenes ISO.

Tabla 17. Requisito software funcional 03.

Requisito Software Funcional	RSF-04
Descripción	Para la gestión se proveerá de un sistema de copia instantánea de máquinas que permitirá la realización de una "foto" de la máquina en cualquier momento y una recuperación instantánea de la misma.

Tabla 18. Requisito software funcional 04.

Requisito Software Funcional	RSF-05
Descripción	Para la creación de los recursos software específicos deberá disponerse de una herramienta que permita instanciar nodos con recursos software especificados seleccionados por el usuario.

Tabla 19. Requisito software funcional 05.

Requisito Software Funcional	RSF-06
Descripción	Para la creación de los recursos hardware específicos se dispondrá de una herramienta de gestión del entorno virtual. Dicha herramienta permite instanciar nodos con recursos hardware especificados a partir de las plantillas disponibles.

Tabla 20. Requisito software funcional 06.

Requisito Software Funcional	RSF-07
Descripción	Los recursos hardware que se pueden asignar a cada una de las instancias son: CPU, memoria RAM, y disco duro.

Tabla 21. Requisito software funcional 07.

Requisito Software Funcional	RSF-08
Descripción	Se facilitará al usuario, de forma sencilla e intuitiva, la definición de distintas VLAN independientes sobre las que se levantarán las máquinas virtuales.

Tabla 22. Requisito software funcional 08.

Requisito Software Funcional	RSF-09
Descripción	El sistema debe contar con un servidor DHCP para la gestión del tráfico entre las distintas VLAN.

Tabla 23. Requisito software funcional 09.

Requisito Software Funcional	RSF-10
Descripción	Para la creación de los distintos entornos de red se generarán una serie de puntos de red que apunten a los entornos determinados.

Tabla 24. Requisito software funcional 10.

Requisito Software Funcional	RSF-11
Descripción	Para poder crear un entorno aislado el sistema debe disponer de la capacidad de generar entornos de red internas dentro del sistema para el aislamiento externo en pruebas concretas.

Tabla 25. Requisito software funcional 11.

Requisito Software Funcional	RSF-12
Descripción	Habrá un único usuario Administrador para la gestión de usuarios y administración de las máquinas.

Tabla 26. Requisito software funcional 12.

Requisito Software Funcional	RSF-13
Descripción	El acceso al sistema se realiza únicamente desde un equipo “front-end” que dispondrá de un entorno gráfico en el que se solicite las credenciales de acceso y se asignen los permisos necesarios sobre los distintos recursos. Adicionalmente, se deberá disponer de un acceso mediante ssh para la conexión a las máquinas Linux.

Tabla 27. Requisito software funcional 12.

Por su parte, los requisitos *software no funcionales* son los siguientes.

Requisito Software no Funcional	RSNF-01
Descripción	El sistema tendrá la capacidad de proveer de un sistema de alta disponibilidad del entorno.

Tabla 28. Requisito software no funcional 01.

Requisito Software no Funcional	RSNF-02
Descripción	El sistema debe ser capaz de repartir la carga de cada una de las instancias entre varios servidores físicos.

Tabla 29. Requisito software no funcional 02.

Requisito Software no Funcional	RSNF-03
Descripción	El sistema estará provisto de una herramienta de administración para la creación y destrucción de máquinas virtuales.

Tabla 30. Requisito software no funcional 03.

Requisito Software no Funcional	RSNF-04
Descripción	El sistema ofrecerá la seguridad necesaria mediante la creación de redes privadas y aisladas.

Tabla 31. Requisito software no funcional 04.

Requisito Software no Funcional	RSNF-05
Descripción	El sistema ofrecerá alta disponibilidad mediante la redundancia cíclica de discos (RAID).

Tabla 32. Requisito software no funcional 05.

Requisito Software no Funcional	RSNF-06
Descripción	El sistema ofrecerá alta disponibilidad mediante copias de respaldo de los datos.

Tabla 33. Requisito software no funcional 06.

2.9. Requisitos hardware

Requisito Hardware	HR-01
Descripción	Los servidores deben contar con la tecnología de virtualización Intel-VT o AMD-V.

Tabla 34. Requisito hardware 01.

Requisito Hardware	HR-02
Descripción	Los servidores deben contar con la capacidad de cómputo suficiente para albergar simultáneamente varias máquinas virtuales interconectadas.

Tabla 35. Requisito hardware 02.

Requisito Hardware	HR-03
Descripción	Los servidores deben contar con conexión a Internet.

Tabla 36. Requisito hardware 03.

Requisito Hardware	HR-04
Descripción	Los servidores deben contar con un dispositivo gráfico para poderse conectar a él.

Tabla 37. Requisito hardware 04.

Requisito Hardware	HR-05
Descripción	La infraestructura debe contar con un sistema de almacenamiento que permita el uso del protocolo NFS.

Tabla 38. Requisito hardware 05.

Requisito Hardware	HR-06
Descripción	Los servidores deberán disponer de al menos dos tarjetas de red para separar los tráficos de red..

Tabla 39. Requisito hardware 06.

Capítulo 3

3. DISEÑO

De acuerdo con el esquema del proyecto, en el Capítulo 3 se muestra la arquitectura global del sistema y más en detalle el diseño de la arquitectura de los laboratorios creados. Además, en el mismo se exponen diagramas del funcionamiento de los mismos.

3.1. Arquitectura global

Previamente al análisis del diseño de los laboratorios de forma individual, en esta sección, se explica la arquitectura general que se ha definido para la realización de los diferentes entornos.

La configuración llevada a cabo se fundamenta en el aislamiento de la red de datos y de gestión de la red pública o Internet, evitando así posibles intrusiones en los datos sensibles y aumentando la eficiencia del sistema. Por ello, los hipervisores tendrán 2 interfaces de red conectadas al *switch* de comunicaciones del Departamento y el almacenamiento dispondrá únicamente de un interfaz de red activo conectado a la red de datos. A continuación se puede apreciar la arquitectura global del sistema:

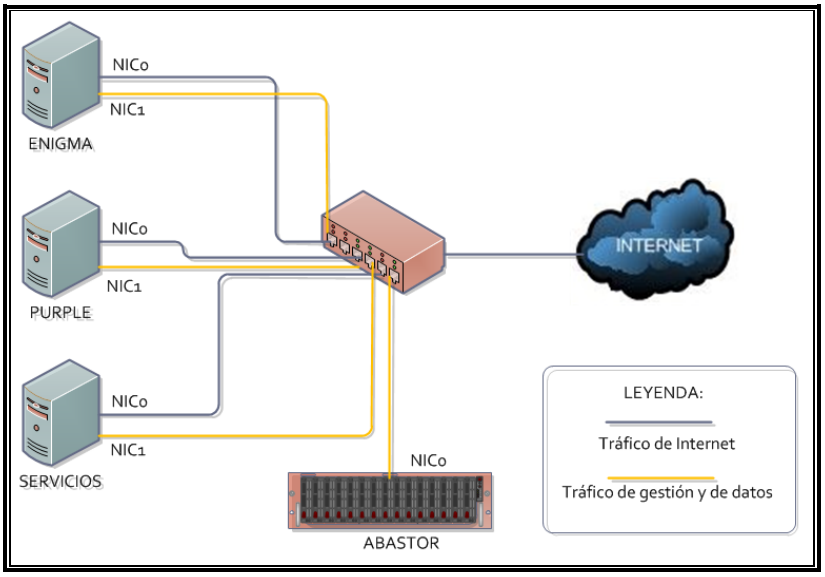


Figura 33. Arquitectura general del entorno.

Como se puede apreciar en la *Figura 33. Arquitectura general del entorno*., las interfaces de red 0 (NIC_0) de los hipervisores irán conectadas a la red pública o de Internet y las interfaces de red 1 (NIC_1) irán conectadas a la red privada o de datos.

Para llevar a cabo la separación de los diferentes tráficos de red, fue necesario crear una nueva red virtual o VLAN en el *switch*. Esta nueva red, a partir de ahora denominada VLAN2, será la red privada o de datos del entorno y no dispondrá de salida fuera del *switch*. La red pública o de Internet, que se denominará en adelante como VLAN1, sí tendrá salida al exterior. La *Figura 34. Vista switch comunicaciones.* resume el esquema final del *switch* de comunicaciones en relación a las VLANs comentadas:

Figura 34. Vista *switch* comunicaciones.

VLAN1: Bocas 4, 5 y 6.

VLAN2: Bocas 17,18, 19 y 20.

Las ventajas e inconvenientes de este diseño son las siguientes:

- Seguridad del sistema. Los datos del servidor de almacenamiento y la gestión de los hipervisores quedan aislados en una red interna.
- Eficiencia. Los datos de las máquinas virtuales generan mucho tráfico de red al estar éstos en el servidor de almacenamiento. La separación de las redes, evita que la red se sature y que los datos salgan del *switch*, evitando un mayor número de saltos de comunicación innecesarios.
- Gestión de los servidores. Por el contrario, la accesibilidad a los servidores se ve limitada al estar en una red aislada. Sin embargo, para minimizar este impacto, se crea una máquina que tiene una tarjeta de red en la red pública y otra en la red privada, donde los servidores sí son accesibles.

En base a esta arquitectura global, se desarrollan dos laboratorios independientes para la realización de pruebas por parte del Departamento:

- Laboratorio automático: Este laboratorio proporciona un entorno virtual para llevar a cabo experimentos automatizados de ejecución de malware.
- Laboratorio web: En este otro entorno, se proporciona un entorno web para la creación de *nubes* privadas.

3.2. Laboratorio Automático

En primer lugar, en este apartado se explica la arquitectura utilizada en el diseño del laboratorio automático, definiendo los aspectos más relevantes del mismo. A grandes rasgos, el objetivo del laboratorio automático consiste en crear un entorno de ejecución malware de forma mecánica a través de la ejecución de un programa que ejecute los comandos oportunos. A modo de resumen, el laboratorio automático debe realizar los siguientes pasos:

- Crear una red virtual aislada.
- Implantar una serie de máquinas virtuales en la misma.
- Desplegar un programa de *malware* en dicho entorno virtual.
- Establecer un tiempo de funcionamiento del mismo.
- Extraer trazas para su posterior estudio.
- Destrucción del entorno generado.
- Analizar las trazas extraídas, con el objetivo de obtener unas conclusiones que faciliten una futura identificación de vulnerabilidades.

La arquitectura utilizada y la secuencia de pasos específica que se establecen para su utilización se muestran a continuación.

3.2.1. Arquitectura

En esta sección se describe la arquitectura que se utiliza en el diseño de la generación del entorno aislado donde se realizan las pruebas con el malware correspondiente.

La arquitectura base está formada por las siguientes máquinas:

- **Máquina anfitriona**: Se denomina así a la máquina que proporciona el soporte hardware para el laboratorio. Sobre el hardware de la máquina se instala un hipervisor que permite la gestión de las máquinas virtuales. El hipervisor utilizado es XEN.
- **Consola de gestión:** Se trata de una máquina virtual que aloja el programa que gestiona el laboratorio de forma autónoma y automatizada. Además, en ella se guardan los ficheros de configuración del experimento y de las máquinas virtuales, así como los resultados de la ejecución del experimento.
- **Servidor DHCP:** Alojado en otra máquina virtual, es el encargado de proporcionar direccionamiento IP al entorno.
- **Servidor OSSIM:** Es el equipo virtual donde está instalado el software de monitorización del entorno. En él se recogen las trazas oportunas que posteriormente serán analizadas.
- **Máquinas virtuales atacantes:** Se generan un número determinado de máquinas virtuales que pueden ejecutar un malware determinado. A su vez,

estas máquinas pueden ser también consideradas víctimas al presentar vulnerabilidades. Llevan instalado el agente OSSEC.

- **Máquinas virtuales víctimas:** Es necesario desplegar un conjunto de máquinas virtuales con distintas vulnerabilidades presentes. Llevan instalado el agente OSSEC.

Inicialmente, las máquinas atacantes/víctimas son máquinas con Windows como sistema operativo. Como se comenta en la sección *2.6*, el entorno de monitorización es capaz no solo de monitorizar sistemas Windows, sino también otros sistemas como Linux, Android, etc, pudiendo existir un entorno mixto.

Las distintas máquinas que forman el laboratorio están conectadas entre sí por una red virtual privada, sin salida a Internet, que genera el propio hipervisor bajo demanda. A su vez, la red virtual privada está definida por un servidor DHCP, que se encarga de ofertar las correspondientes direcciones IP a los clientes. Las redes virtuales privadas no disponen de salida a Internet por motivos de seguridad ya que pueden suponer un riesgo innecesario en la red. Como línea de acción futura, se podría estudiar la posibilidad de establecer un servidor proxy capaz de permitir la salida a Internet de las máquinas, sin resultar un riesgo potencial para la red.

También es posible crear tantos experimentos, aislados entre sí, como se desee. La única limitación está determinada por los recursos físicos (CPU, RAM…) del hipervisor. Los diferentes proyectos se generan desde la consola, que constituye el único punto de unión entre los proyectos. Cada experimento que se lanza, se ejecutan las siguientes tres fases:

- En primer lugar, se configura el entorno y se definen las diferentes máquinas virtuales que van a intervenir en el experimento, así como el *software* a incluir en cada máquina. También en esta fase se van a determinar las máquinas atacantes y las víctimas.
- Posteriormente, en una segunda fase, se procede a desconectar la consola de la red virtual creada y se comienza a ejecutar el experimento.
- En la última fase, se eliminan las máquinas virtuales, previa recopilación de la información almacenada en el servidor de detección (OSSIM).

La *Figura 35. Esquema genérico del laboratorio malware*. muestra de forma gráfica la arquitectura a implementar:

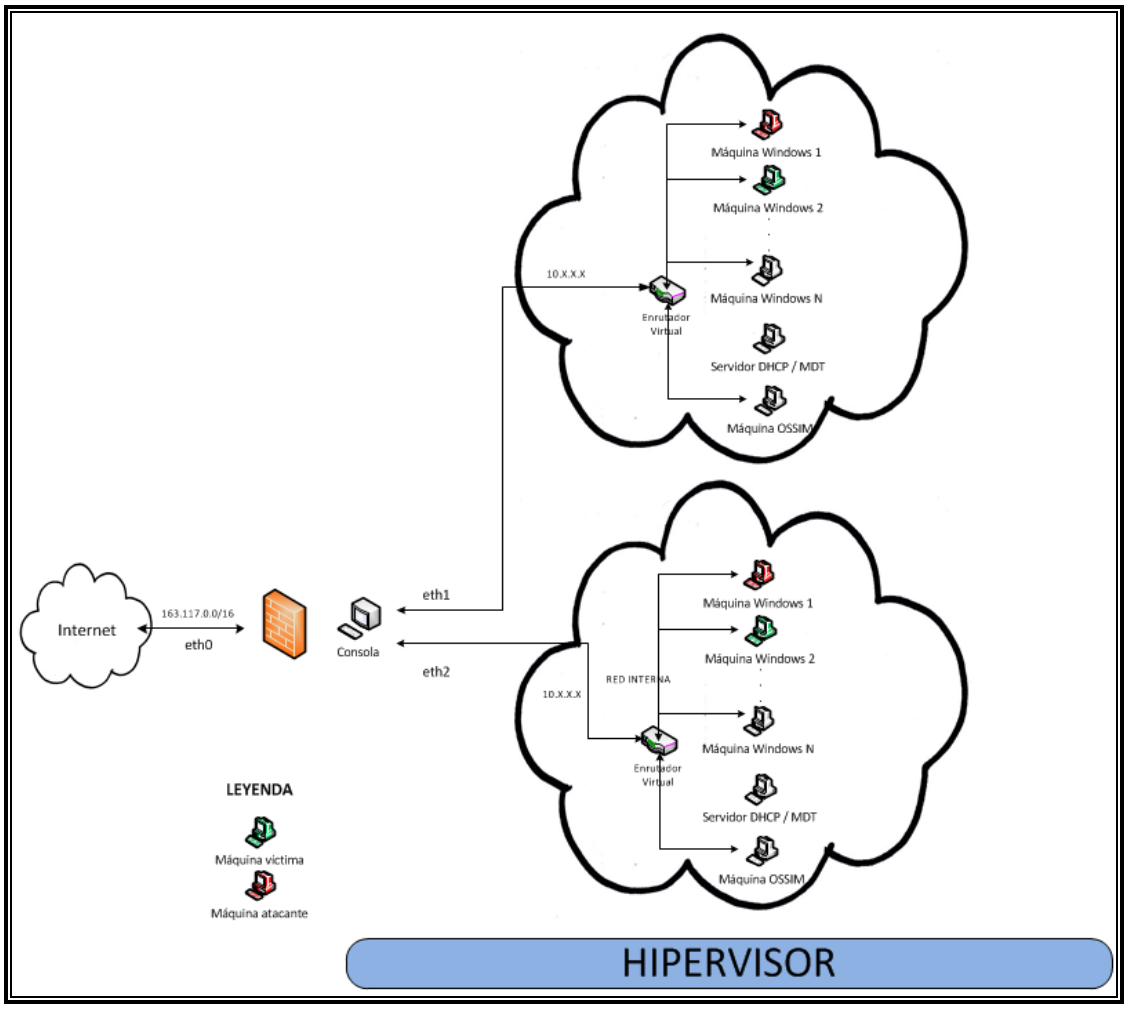


Figura 35. Esquema genérico del laboratorio malware.

3.2.2. *Definición de las máquinas virtuales*

Uno de los objetivos de este entorno consiste en obtener la mayor información posible del impacto que un *malware* puede provocar en un entorno aislado de pruebas. Por esta razón, interesa diseñar un entorno flexible para dar cabida al mayor número posible de sistemas operativos, con el mayor número posible de combinaciones de aplicaciones a instalar.

3.2.2.1. *Definición de la consola de gestión.*

La consola de gestión es la máquina que gestiona todas las operaciones para la implantación de los diferentes experimentos. En ella se ejecutan los *scripts* de generación de los experimentos, se realizan las configuraciones oportunas y se definen las diferentes máquinas virtuales. La definición de las máquinas virtuales puede, bien estar definida por el usuario, o bien ser aleatoria, en todos los sentidos. Será aleatorio tanto el número de máquinas que se creen como el *software* que se instale en cada una de ellas. Cabe destacar que la importancia de la consola radica en la primera fase de generación del laboratorio, donde realiza las tareas anteriormente comentadas. Una vez esté definido el laboratorio, se *desconecta* la consola del experimento, aislando así a la

consola del entorno de ejecución del malware. La propia consola tiene mecanismos para saber cuándo las máquinas han sido totalmente configuradas, que se detallan en el Apartado *3.2.3.3* de esta memoria. Cuando están todas máquinas configuradas, se procede a deshabilitar la tarjeta de red y a enviar una señal, en forma de fichero, a las máquinas virtuales.

Cabe destacar que el acceso a la consola ha sido restringido con el fin de evitar accesos no autorizados al laboratorio.

Como consola de gestión, se ha considerado conveniente instalar, como sistema operativo, la versión Precise de Ubuntu (Ubuntu 12.04.1 LTS). Se trata de la última versión disponible, en la actualidad, de su sistema operativo que, además, es gratuito y sencillo de utilizar. Además, se precisa configurar una máquina que sea capaz de interpretar comandos Xen para la creación y configuración del laboratorio. Dado que Ubuntu posee una serie de *paquetes* que permiten la interactuación con Xen a través de su API de programación, se ha decidido instalar esta versión de Ubuntu. Aparte del sistema operativo, esta máquina dispone de los programas que se deberán ejecutar para crear y configurar el laboratorio, así como el repositorio de aplicaciones.

La consola, como se puede apreciar en la *Figura 35. Esquema genérico del laboratorio malware.*, dispone de una interfaz de red pública conectada a la red del Departamento (red 163.117.149.0/25), que es accesible únicamente desde la misma, mediante la configuración de las reglas de cortafuegos correspondientes. Adicionalmente, dispone de tantas interfaces de red privadas (generadas internamente en el servidor Xen), sin salida a Internet, como experimentos se estén ejecutando simultáneamente. Se crea una red privada por experimento, estando las redes aisladas entre sí, lo que evitará posibles *contaminaciones* entre los distintos experimentos.

3.2.2.2. Definición de máquinas atacantes / victimas.

Las máquinas virtuales generadas en los experimentos se generan automáticamente, siguiendo el mismo procedimiento, ya se trate del supuesto de un equipo *atacante* o del de una *víctima*. La máquina se crea, se configura y, en último lugar, se establece si va a ejecutar una pieza de *malware* o no. Esta decisión se adopta en base a un algoritmo por el cual, al menos la mitad de las máquinas virtuales de cada tipo van a simular ser equipos comprometidos con malware (en adelante máquinas atacantes). Esto es, la única diferencia que existe entre máquinas atacantes y víctimas es la ejecución, o no, de una pieza de malware.

Para la elección de las máquinas virtuales atacantes o víctimas, se ha decidido instalar Windows (pudiendo ser también máquinas con Linux como sistema operativo) como sistema operativo, precisamente porque se trata de un sistema del cual se conocen muchas más vulnerabilidades y que tiene una enorme presencia mundial como se desprende de la siguiente gráfica:

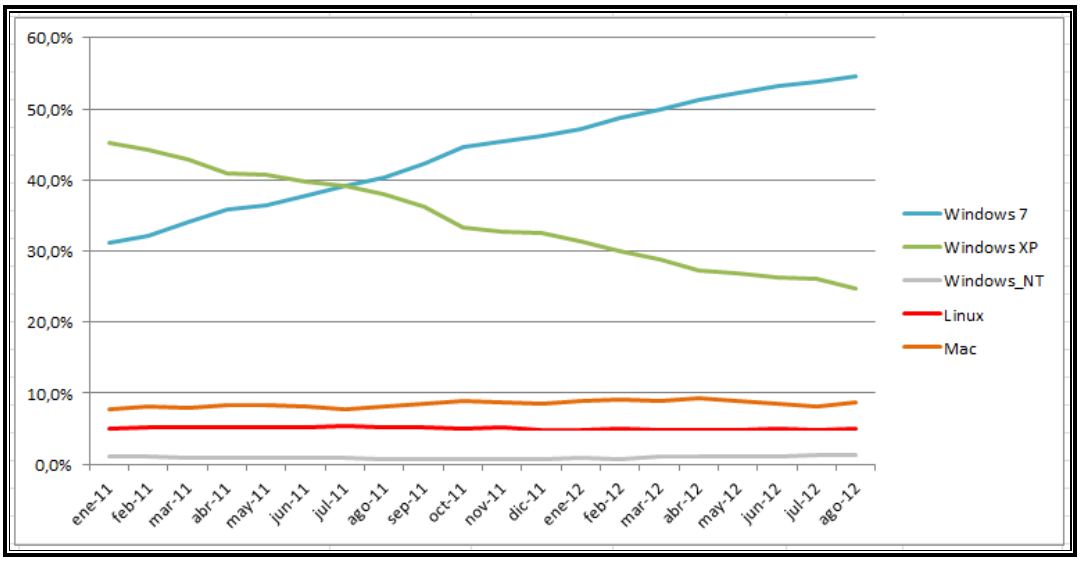


Figura 36. Evolución del uso de sistemas operativos por mes.[30]

La lista definitiva de sistemas operativos utilizados incluye a toda la gama de sistemas operativos Windows posibles, incluyendo el último sistema operativo que acaba de salir al mercado: Windows 8. La lista definitiva de los sistemas operativos seleccionados, se muestra en la siguiente tabla:

Sistemas operativos disponibles
Windows XP
Windows XP SP1
Windows XP SP2
Windows XP SP3
Windows 7 SP1
Windows 2003 SP2 x64
Windows 2008 R2 x64
Windows 8

Tabla 40. Sistemas operativos utilizados en el laboratorio automático.

De esta forma, se dispone de una amplia variedad de versiones de Windows donde poder realizar un análisis de los ataques malware que se deseen ejecutar.

Se pretende, por tanto, monitorizar todas y cada una de las máquinas del ejercicio. Para ello, cada máquina tiene instalado de base el agente de OSSEC para la monitorización del sistema. El fichero de configuración y los datos a monitorizar se proporcionan en la creación del entorno aislado.

Asimismo, ha sido necesario seleccionar una serie de aplicaciones que, instaladas en estos sistemas operativos, permitan establecer un laboratorio con las suficientes variables para efectuar varios experimentos. Las características de las aplicaciones son:

- Necesidad de diversidad en el entorno para el análisis de las vulnerabilidades.
- La instalación se efectúa de forma automática y desatendida.

[30] http://www.w3schools.com/browsers/browsers_os.asp

- Su puesta en marcha es sencilla ya que las aplicaciones se incluyen en un fichero.

Estas aplicaciones han sido seleccionadas en base a un informe de seguridad en el que se enumeran las aplicaciones de Microsoft o de terceros más vulnerables del mercado.[31] Adicionalmente, se han añadido otras aplicaciones seleccionadas en base a su gran popularidad y alta utilización diaria. La lista de aplicaciones a poder instalar en cada una de las máquinas virtuales (ya sean atacantes o víctimas) se muestra en la siguiente tabla:

Lista de aplicaciones homologadas
Adobe Reader 9.1
Adobe Reader 10.1.0
Adobe Flash Player 9.0.124.0
Adobe Flash Player 10.0.32.18
Adobe Flash Player 10.3.183.7
Mozilla Firefox 6.0.2
Mozilla Firefox 13.0.1
K-Lite Codec pack 5.1.0
K-Lite Codec pack 9.2.0
.NET Framework 1.1 SP1
.NET Framework 2.0 SP2
.NET Framework 3.5 SP1
Java 1.4.2.07
Java 1.5.0.11
Java 1.6.0.70
MSXML 4 SP2
MSXML 6
QuickTime 6.5
QuickTime 7.64.17.73
Macromedia Shockwave 9
Macromedia Shockwave 11.0
Google Chrome 21.0.1180.89

Tabla 41. Lista de aplicaciones utilizadas en el laboratorio automático.

Tanto los sistemas operativos utilizados, como las aplicaciones descritas, son de carácter gratuito. Las aplicaciones se pueden descargar libremente a través de Internet, mientras que los sistemas operativos se han descargado del enlace que el Departamento de Informática de la Universidad Carlos III de Madrid mantiene con el MSDNAA (*Microsoft Developer Network Academic Alliance*). Gracias a los acuerdos que la Universidad tiene con Microsoft, la descarga y utilización de los sistemas operativos utilizados en el laboratorio han sido gratuitas.

En la *Tabla 42. Resumen de la homologación de las aplicaciones en los sistemas operativos*., por un lado, se indica la homologación de las aplicaciones en los distintos

[31] Secunia Half Year Report 2010. http://secunia.com/gfx/pdf/Secunia_Half_Year_Report_2010.pdf

sistemas operativos y, por otro, se establecen las limitaciones oportunas para que dos paquetes no se instalen simultáneamente en un equipo. Esto último implica que, por ejemplo, no se pueden instalar en un mismo equipo dos versiones diferentes de Adobe Reader. Las aplicaciones resaltadas con el mismo color serán excluyentes (esto es, en cuanto se instale una de ellas la otra no se instalará).

Aplicación	Windows XP SP2	Windows XP SP3	Windows 2003 x64	Windows 7	Windows 2008
Adobe Reader 9.1	□	□	□	□	□
Adobe Reader 10.1.0	□	□	□	□	□
Adobe Flash Player 9.0.124.0	□	□	□	□	□
Adobe Flash Player 10.0.32.18	□	□	□	□	□
Adobe Flash Player 10.3.183.7	□	□	□	□	□
Mozilla Firefox 6.0.2	□	□	□	□	□
Mozilla Firefox 13.0.1	□	□	□	□	□
K-Lite Codec pack 5.1.0	□	□	□	□	□
K-Lite Codec pack 9.2.0	□	□	□	□	□
.NET Framework 1.1 SP1	□	□	□	□	□
.NET Framework 2.0 SP2	□	□	□	X	X
.NET Framework 3.5 SP1	□	□	□	X	X
Java 1.4.2.07	□	□	□	□	□
Java 1.5.0.11	□	□	□	□	□
Java 1.6.0.70	□	□	□	□	□
MSXML 4 SP2	□	□	□	□	□
MSXML 6	□	□	□	X	X
QuickTime 6.5	□	□	□	□	□
QuickTime 7.64.17.73	□	□	□	□	□
Macromedia Shockwave 9	□	□	□	□	□
Macromedia Shockwave 11.0	□	□	□	□	□
Google Chrome 21.0.1180.89	□	□	□	□	□

Tabla 42. Resumen de la homologación de las aplicaciones en los sistemas operativos.

3.2.2.3. Definición de la máquina DHCP

Esta máquina es la encargada de definir y gestionar la red virtual privada. Cuando la consola genera la red, ésta se crea, pero no queda definida. Los clientes, al no disponer de una definición de red, establecen, al cabo de varios minutos, la configuración por defecto de Windows. Para minimizar ese tiempo, se establece un servidor DHCP que se encarga de definir la red y asignar direcciones IPs a los clientes.

Como selección de máquina para realizar las funciones de DHCP (*Dynamic Host Configuration Protocol)* se ha optado por utilizar un Windows 2008 R2 64 bits. El rol de DHCP va a consistir en ofrecer direcciones IP a las máquinas virtuales que se inicien en el laboratorio, salvo la de dirección IP de la interfaz de red de la consola de administración en el laboratorio, que dispone de IP fija fuera del rango de DHCP y la dirección IP de la máquina que actúa como detectora de las posibles vulnerabilidades. Su IP está configurada de forma estática en la 10.10.10.1 y el rango de direcciones IP que ofrece es de la IP 10.10.10.2 a la IP 10.10.10.100 con una máscara de red 255.255.255.0.

3.2.2.4. Definición de la máquina de control: OSSIM

Como máquina para la detección de las intrusiones en los sistemas, se ha seleccionado una herramienta de código abierto: OSSIM. Para su instalación se utiliza una imagen previamente descargada de Internet que posee todas las características necesarias. Simplemente es necesario configurar el equipo para satisfacer las necesidades del entorno de ejecución. Al seleccionar como sistema de monitorización el agente de OSSEC, es necesario configurar el *plugin* de OSSEC en el servidor, para que así pueda interpretar la información de los agentes. La versión utilizada de OSSIM será la versión 4.1 del producto.

3.2.2.5. Definición de la máquina huésped o Hipervisor

El hipervisor es, como se ha explicado anteriormente, la capa que media entre los huéspedes o máquinas virtuales y el hardware físico del servidor. Se utiliza como capa de virtualización Xen, en este caso su versión gratuita, XCP v1.5 (Xen Cloud Platform). Sobre el hipervisor se generan las redes virtuales privadas y las máquinas virtuales. Sobre esta capa se desarrolla la totalidad de los experimentos, proporcionando los recursos necesarios para que se puedan ejecutar correctamente.

3.2.3. Descripción detallada del sistema

Dentro de este apartado se desarrolla en mayor profundidad el funcionamiento del laboratorio automático. Para ello, se especifican las diferentes fases y los procedimientos que se llevan a cabo en cada una de ellas. Las distintas fases del laboratorio son las siguientes:

- Definición del experimento.
- Creación e inicialización del entorno.
- Configuración del entorno.
- Ejecución del experimento.
- Destrucción del entorno.
- Análisis de las trazas generadas.

Estas distintas fases siguen un desarrollo lineal, existiendo relaciones entre ellas. En la siguiente figura se ofrece una visión global de estas diferentes fases, donde se muestran los procedimientos más importantes:

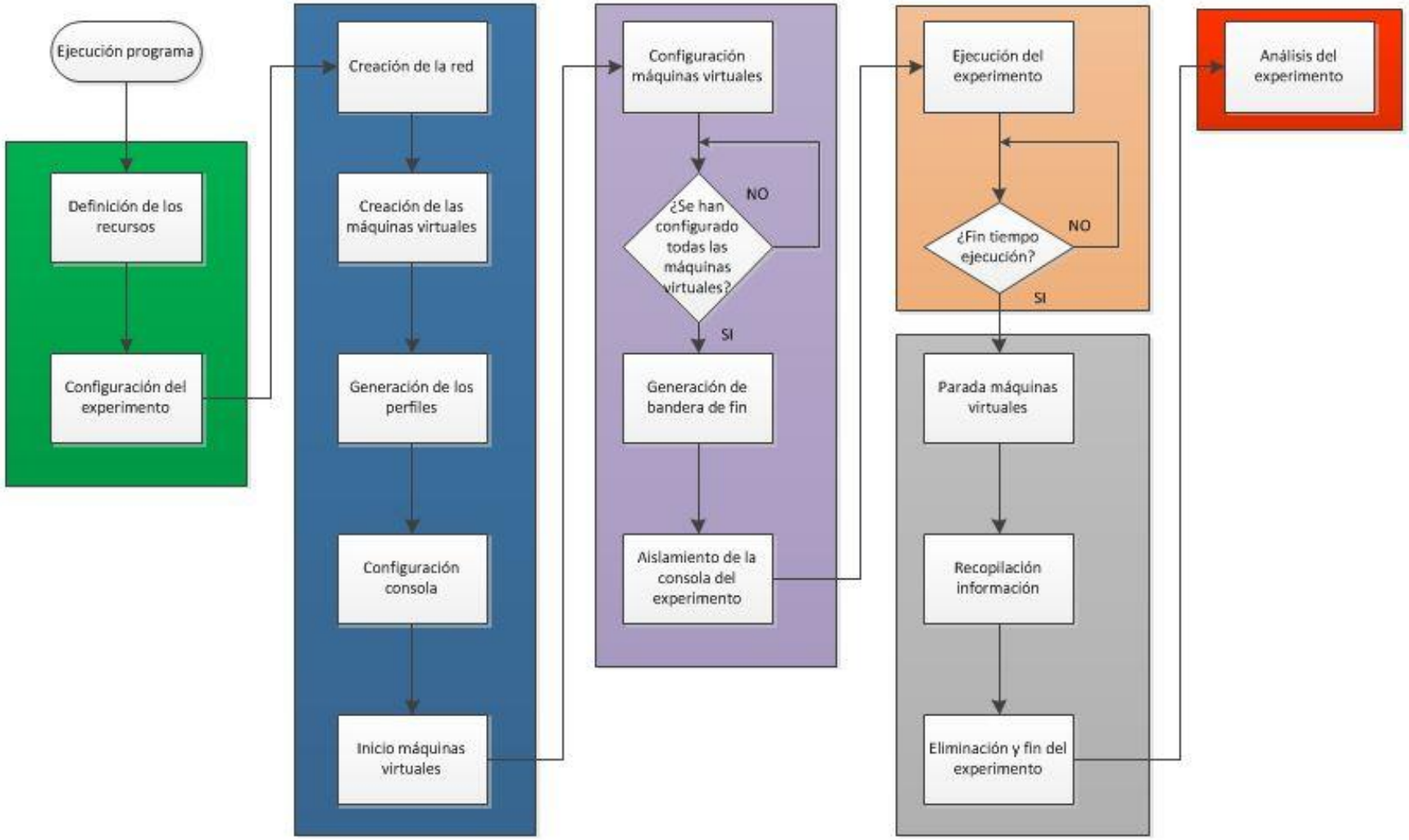


Figura 37. Configuración general del laboratorio automático.

Para una mejor comprensión del proceso, a continuación, se describen las distintas fases que componen el proceso, analizando en profundidad cada una de ellas con la finalidad de tener una visión completa del mismo.

3.2.3.1. Definición del experimento

En esta primera fase se inicializan las variables que se pasan por la línea de comandos, se declaran los sistemas operativos que se van a utilizar durante el experimento y se define el hipervisor donde se va a ejecutar el experimento, se crea un número de experimento en base a la generación de una variable que consta de 14 dígitos generados de forma aleatoria y se crea la estructura de directorios y el fichero de configuración correspondiente. A continuación se muestra el diagrama de flujo de esta primera fase.

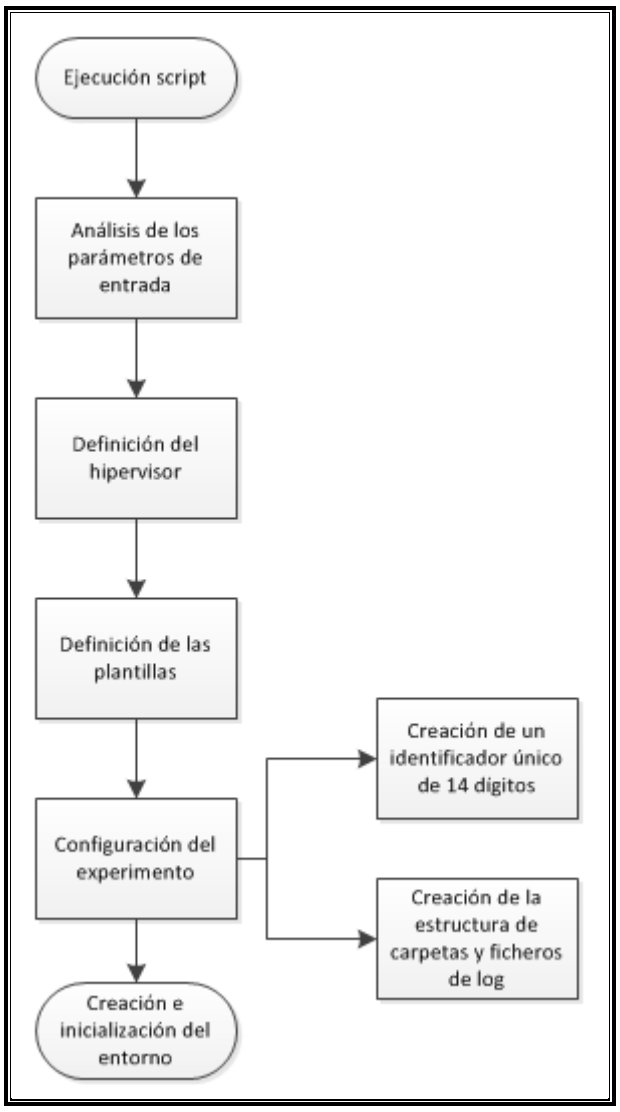


Figura 38. Fase de definición del laboratorio automático.

3.2.3.2. Creación e inicialización del entorno

Esta segunda fase del sistema es, sin lugar a dudas, la más compleja y crítica de la generación del experimento. En la misma fase se crea la red virtual privada, interna al hipervisor, y aislada, tanto de Internet como de otros experimentos. Además, se procederá a inicializar las máquinas de infraestructura que son el servidor DHCP y el servidor OSSIM. El diagrama de esta fase se puede consultar en la *Figura 39. Fase de creación e inicialización del laboratorio automático.*

Asimismo, es en esta fase cuando se generan los perfiles de cada máquina virtual. Estos perfiles definen las aplicaciones que van instaladas en las diferentes máquinas virtuales. Cada perfil se compone de forma aleatoria, seleccionando el número de aplicaciones a instalar y escogiendo las mismas entre las existentes en un fichero donde se encuentran todas las aplicaciones disponibles. Una vez generados los perfiles y las máquinas virtuales, se les asigna la red anteriormente creada. Posteriormente se asigna y se activa la tarjeta de red en la consola, pues va a ser en ésta donde se encuentre el repositorio de aplicaciones y los ficheros que posteriormente deberán ejecutarse en las máquinas virtuales. Con el fin de permitir la generación de varios laboratorios de forma simultánea, será necesario establecer una lógica en el programa que permita establecer un interfaz de red libre en la consola de administración. Una vez se haya determinado el interfaz libre, se le asignará la red virtual privada y se activará, asignándole la IP 10.10.10.101 y una máscara de red 255.255.255.0. Se ha restringido la creación de experimentos a 3 ya que los recursos del laboratorio son limitados.

Por último, se inician las máquinas virtuales definidas y comienza la fase de configuración del entorno.

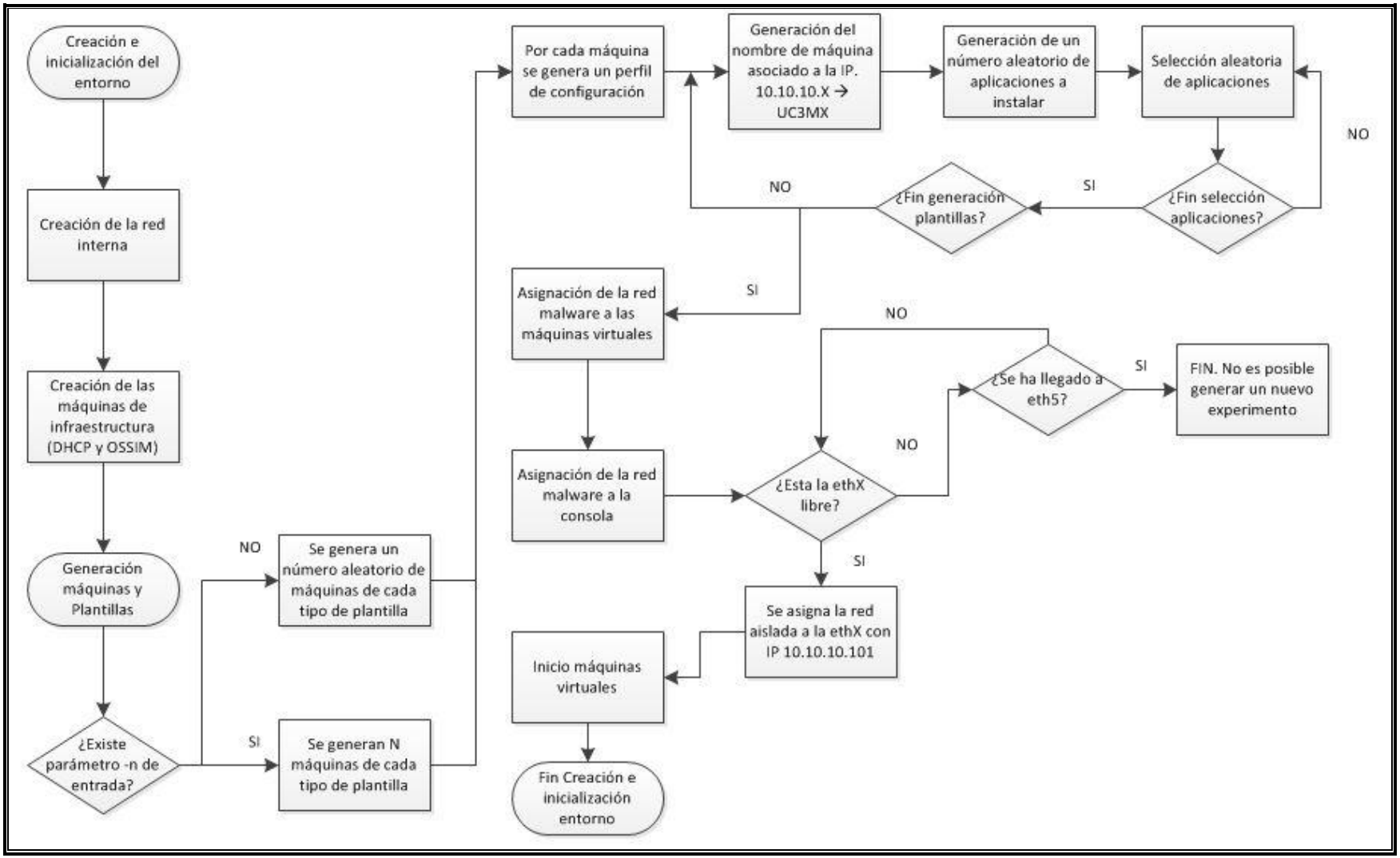


Figura 39. Fase de creación e inicialización del laboratorio automático.

3.2.3.3. Configuración del entorno

Una vez iniciadas las máquinas virtuales, comienzan a ejecutarse dos procesos paralelos: el primero de ellos, tiene lugar en la consola de gestión. Consiste en un bucle que espera a que todas las máquinas virtuales que se han creado estén totalmente configuradas.

El segundo proceso se lleva a cabo en cada máquina virtual de forma simultánea. Éste es el proceso en el que se configura cada máquina virtual. Dicho proceso debe ser totalmente independiente y autónomo. Por ello, se define en cada plantilla un fichero genérico que se ejecuta en el inicio de la máquina. Este fichero es el encargado de copiar los ficheros necesarios, que están centralizados en la consola, para su posterior ejecución. De esta forma, se evita que haya que estar modificando las plantillas en el caso de tener que modificar un fichero y se dispone de un punto central de configuración.

Una vez sincronizadas las aplicaciones a instalar y los ficheros de ejecución, se procede a sincronizar el perfil de la máquina virtual. Este fichero, contiene el nuevo nombre de máquina, el nombre del ejecutable del malware (en el caso de que le corresponda su ejecución) y las aplicaciones que le corresponde instalar. Tras cambiar el nombre al equipo, éste se reinicia.

Después del reinicio, se pasa a ejecutar el programa que instala, de forma desatendida, las aplicaciones que se indican en el perfil del equipo. Cuando el equipo está totalmente configurado se envía una notificación, en forma de fichero, a la consola para indicar que está configurada.

Una vez han finalizado de configurarse todas las máquinas virtuales, se envía una bandera desde la consola a éstas para que se reinicien. En ese momento, se deshabilita la tarjeta de red de la consola asignada a este experimento para evitar una posible contaminación de la misma.

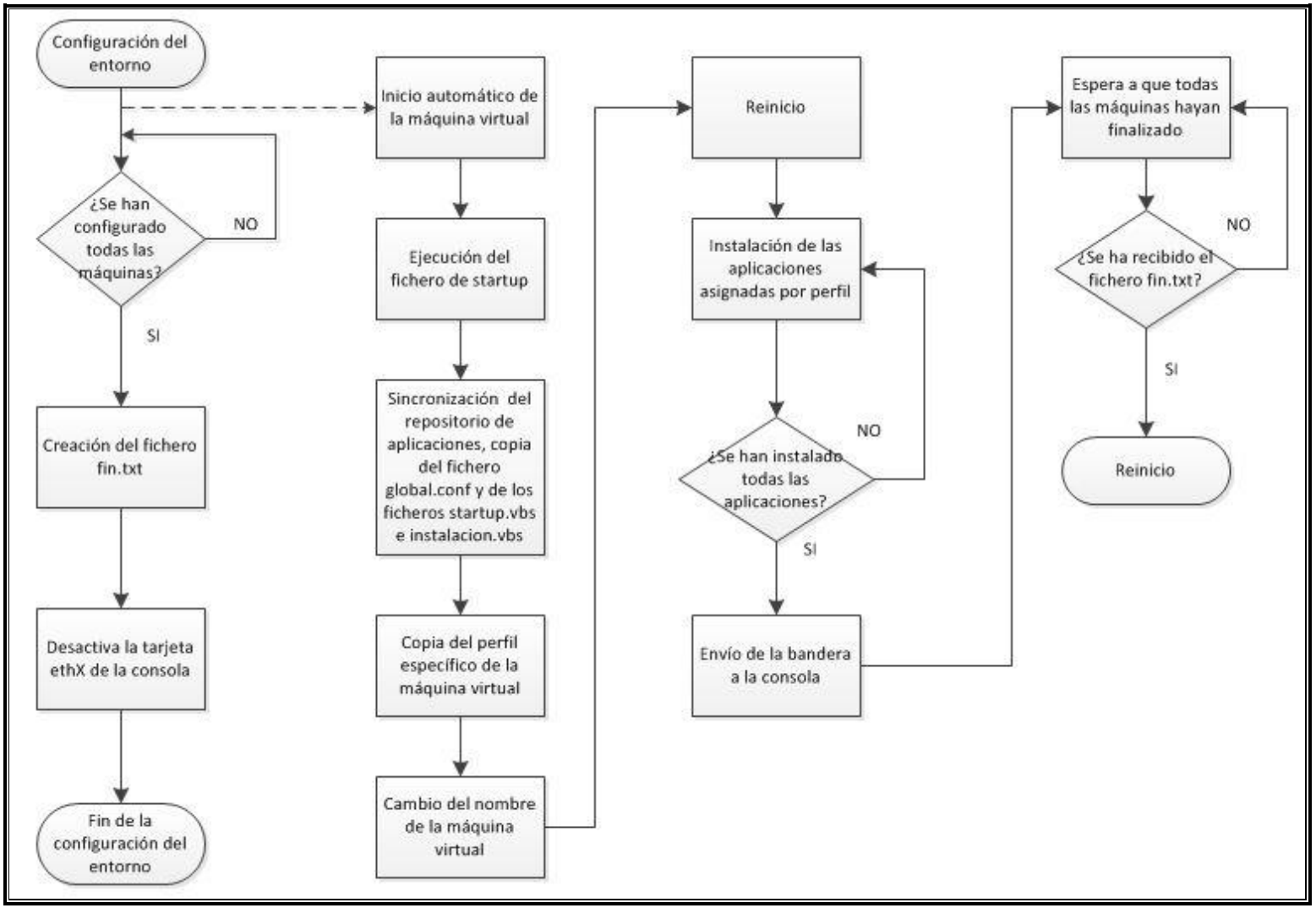


Figura 40. Fase de configuración del laboratorio automático.

3.2.3.4. Ejecución del experimento

Es en esta cuarta fase cuando se procede a ejecutar el *malware* en las máquinas que así lo tengan definido en su perfil. El laboratorio permanece en funcionamiento el tiempo que se haya definido en la ejecución del programa. Durante este periodo de tiempo, el laboratorio malware permanece aislado, ejecutándose de forma autónoma.

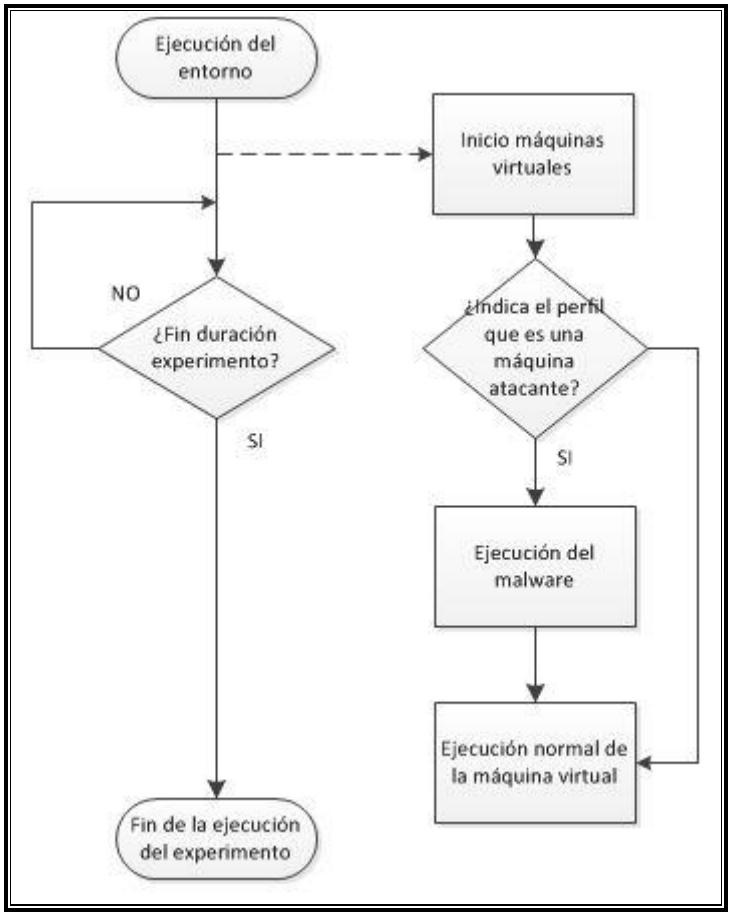


Figura 41. Fase de ejecución del laboratorio automático.

3.2.3.5. Destrucción del entorno

Finalizado el tiempo de ejecución del laboratorio, se procede a la eliminación del entorno creado. En este punto se apagan las máquinas virtuales que han formado parte del laboratorio y, una vez apagadas, se elimina el disco virtual asignado a cada una de ellas para, posteriormente eliminar la máquina virtual en sí. A continuación, se apaga y elimina el servidor DHCP. Igualmente, una vez recopilada del servidor OSSIM los ficheros necesarios, se apaga y elimina el servidor OSSIM. Además, en la consola de gestión se elimina la tarjeta de red asignada al experimento, dejando libre ese interfaz de red para futuros experimentos. Por último, se elimina la red virtual creada al inicio del experimento. Hay que resaltar que, si no se apagan y eliminan todos los elementos previamente, la red no se puede eliminar al tener definidos punteros a los diferentes elementos.

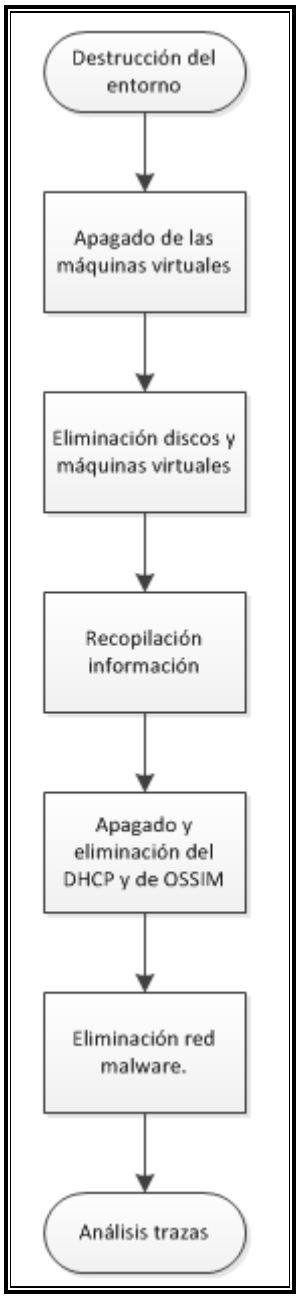


Figura 42. Fase de destrucción del laboratorio automático.

3.2.3.6. Análisis de las trazas generadas

En último lugar, se realiza un estudio de las alertas que se han generado en el servidor OSSIM. Las alertas vienen detalladas en un fichero que se genera en el servidor OSSIM y que se copia a la consola de gestión. El análisis consiste en realizar un examen del fichero realizando un estudio de los diferentes eventos generados como por ejemplo:

- Número total de eventos.
- Número de eventos relacionados con Windows.
- Número de inicios de sesión que se han realizado.
- Cambios en la integridad de ficheros y claves de registro.
- Número de ficheros añadidos.

En el apartado de pruebas se detallan en profundidad el estudio del fichero de eventos.

3.3. Laboratorio Web

Tras haber analizado el diseño del laboratorio automático, en este apartado se explica la arquitectura utilizada en el diseño del laboratorio web, definiendo los aspectos más relevantes del mismo. El objetivo de este entorno es el de proporcionar la creación y administración de nubes privadas en un entorno gráfico amigable en la que los usuarios pueden realizar los experimentos oportunos dentro de un laboratorio virtual. A modo de resumen, el diseño laboratorio web tiene las siguientes fases:

- Establecer un entorno web donde crear instancias de máquinas virtuales.
- Configurar el entorno para proporcionar las características deseadas.
- Crear plantillas base para el despliegue de máquinas virtuales.
- Gestionar un repositorio de imágenes de instalación de sistemas operativos o aplicaciones.

A continuación se analizará la arquitectura utilizada y la configuración específica que se establece para su utilización.

3.3.1. Arquitectura

Esta sección describe la arquitectura diseñada para la generación del entorno web. Se utiliza, para ello, la herramienta **CloudStack**. En la sección *2.5*, ya se ha visto de forma general la arquitectura de este tipo de solución.

En nuestro caso concreto, la arquitectura está formada por cuatro máquinas, más la máquina encargada de realizar la conexión al entorno, que se detallan a continuación:

1. **Acceso Web**: Desde esta máquina se realiza la conexión al laboratorio. No se trata de una única máquina sino que, con este término, se hace referencia al conjunto de máquinas que van a poder acceder al laboratorio de forma simultánea. Para ello, simplemente es necesario que el equipo disponga de un navegador web y que esté conectado a la red de la Universidad Carlos III de Madrid.
2. **Servidor de Gestión o WebFrontEnd**: Será el servidor encargado de realizar todas las operativas del laboratorio, desde crear instancias, redes… hasta la disposición de variables globales de configuración. Este servidor también alojará la base de datos de CloudStack y dispondrá de un servicio web al que se conectará la máquina anterior para interactuar con el laboratorio.

3. **SSHFrontEnd**: Se trata de un punto de acceso al laboratorio mediante consola. Dispone de las configuraciones necesarias para poder realizar una conexión *ssh* con cualquier máquina virtual dentro del laboratorio web.
4. **Hipervisor**: Ésta es la máquina donde se inicien las instancias de las máquinas virtuales. Deberá proporcionar los recursos físicos necesarios para la creación de las máquinas virtuales. El hipervisor utilizado para la realización del entorno web es XenServer en su versión gratuita. Al ser el entorno web un entorno más susceptible a *ataques* que el entorno automático, se ha optado por recurrir a este hipervisor, ya que se trata de un software estable, probado y con actualizaciones de seguridad disponibles en un corto periodo de tiempo, mientras que su versión abierta (Xen Cloud Pltform o XCP) no ofrece este tipo de actualizaciones y la seguridad podría verse comprometida.
5. **Servidores de almacenamiento**: Por último, el laboratorio consta de dos servidores que almacenaran los datos. Por un lado un sistema Abastor[32] donde se reservará un volumen lo suficientemente grande para albergar los discos virtuales del entorno web. Por otro lado un servidor virtual dedicado al almacenamiento secundario con capacidad suficiente para almacenar las imágenes ISO, las plantillas…

En la siguiente figura pueden apreciarse los distintos componentes que forman parte de esta solución web para la creación de *nubes* privadas.

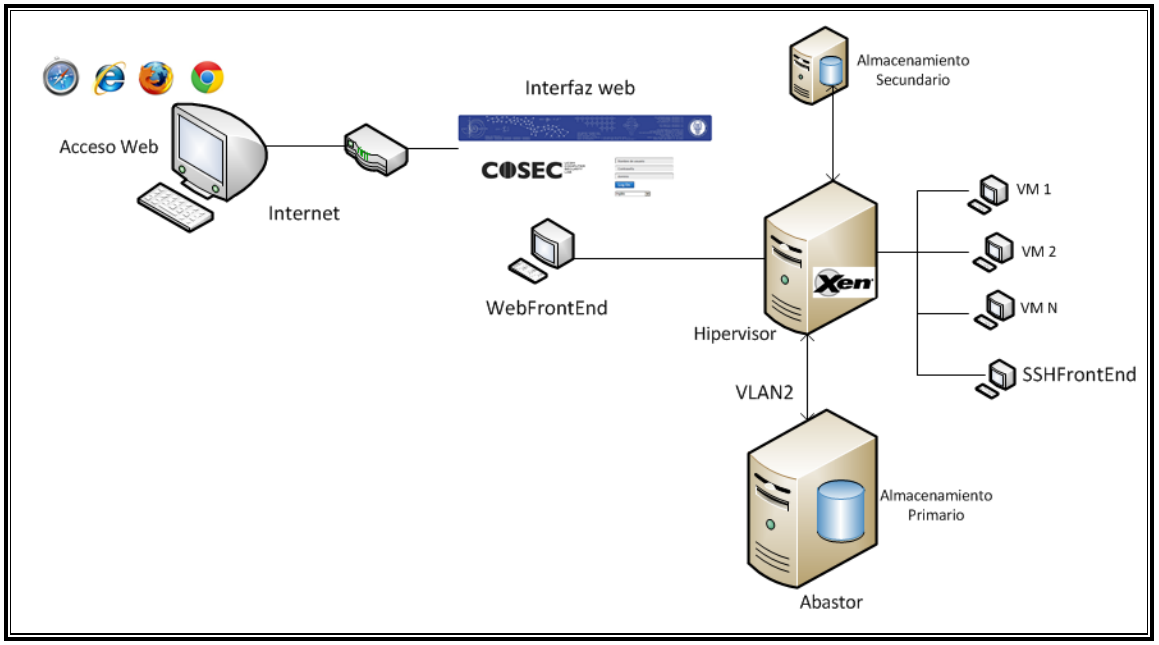


Figura 43. Arquitectura laboratorio web.

[32] http://www.abastor.com/

3.3.2. Definición de las configuraciones

El objetivo de este entorno es el de proporcionar a los usuarios la infraestructura necesaria para el desarrollo de las *nubes* privadas correspondientes. Para que esto sea posible, se han realizado algunas configuraciones en CloudStack para que los usuarios dispongan de los recursos hardware y software necesarios dentro de los requisitos establecidos.

3.3.2.1. Configuración de Plantillas

En primer lugar, se definen un conjunto de plantillas que deben estar disponibles para que los usuarios finales puedan iniciar máquinas virtuales en base a las mismas. En un principio están disponibles 5 plantillas, a saber:

- Ubuntu Server 12.04 LTS
- Ubuntu Desktop 11.10
- Windows XP SP3
- Windows 7
- Windows Server 2008 R2 x64
- Cent OS 5.6

Posteriormente podrán añadirse, si se precisa, tantas plantillas como sean necesarias.

Estas plantillas están configuradas según las configuraciones de seguridad establecidas para cada caso, y con las herramientas y actualizaciones oportunas. Una vez desplegada la máquina virtual, el usuario dispone de la libertad de realizar las modificaciones que considere pertinentes en base a sus necesidades particulares.

3.3.2.2. Configuración de imágenes de arranque

Las imágenes de arranque o imágenes ISO[33], son los ficheros que se utilizan para inicializar e instalar las máquinas virtuales de cero. Se trata de un fichero que contiene todos los *binarios* necesarios para instalar el sistema operativo correspondiente. Será necesaria la configuración de un repositorio de imágenes ISO para así permitir al usuario la instalación de una amplia variedad de sistemas operativos. En un principio no se ofertará al usuario final ninguna imagen ISO ya que el propio usuario tendrá los permisos correspondientes para poder *subir* ficheros al entorno web.

3.3.2.3. Configuraciones globales del sistema

Se han realizado una serie de configuraciones globales que se aplican al entorno y que se describen a continuación:

[33]Imagen ISO ("International Organization for Standardization") es un sistema de ficheros compuesto por el contenido de cada sector de un disco óptico. http://en.wikipedia.org/wiki/ISO_image

- **Configuración https.** Necesaria para definir el acceso al entorno web a través de un protocolo seguro con el correspondiente certificado. Adicionalmente, se realiza una redirección de http a https.
- **Configuración del tiempo de limpieza de máquinas virtuales.** Se dispone de un tiempo de limpieza de máquinas virtuales eliminadas de 12h. Durante las primeras 12h tras eliminar la máquina virtual, es posible su recuperación. Pasadas estas 12h, la máquina con todo su contenido se elimina del entorno. En cualquier caso, no se elimina el disco de datos que se puede utilizar en otra máquina virtual
- **Configuración del envío de alertas.** Con el objetivo de estar informados de los posibles errores en el laboratorio, se configura el envío de alertas por correo electrónico. Para ello, se utiliza el servidor de envío de correos electrónicos de la Universidad y se configuran las direcciones de correo electrónico de los administradores como destinatarios de los mismos.
- **Configuración de alertas.** Se configura una alerta en la que se envía un correo electrónico a los administradores si el entorno está a un 75% de su capacidad de procesamiento o de uso de memoria.
- **Configuración de cuentas y proyectos.** Dado que los recursos iniciales del laboratorio son limitados (sólo se dispone de un servidor con una cuantía determinada de recursos), y para evitar que los recursos no sean correctamente utilizados por los usuarios finales, se delega en el administrador la posibilidad de crear proyectos, cuentas y direcciones IP públicas. El administrador es el único con la capacidad y los permisos necesarios para la creación de proyectos, cuentas y nuevas redes de huéspedes.
- **Configuración de descarga de plantillas e ISOs.** Se configura el sistema para que sea posible descargarse plantillas e imágenes ISO de cualquier dirección de Internet.
- **Configuración del tamaño máximo de disco.** Como medida de control, se limita a 50 GB el tamaño máximo de disco que un usuario es capaz de crear.

3.3.2.4. Configuración de Servicios ofertados

Una de las principales configuraciones que se deben definir es el establecimiento de los servicios que se deseen ofertar a los usuarios. La definición de los mismos debe procurar un equilibrio entra la libertad de decidir de los usuarios en cuanto a sus configuraciones, y los recursos reales disponibles del entorno. Este equilibrio debe favorecer que los recursos se gestionen de forma eficiente y controlada. Hay que tener en cuenta que una falta de control en los recursos puede provocar que la capacidad del sistema se vea mermada por el uso excesivo e innecesario de dichos recursos por parte de los usuarios. En este sentido, los servicios ofertados son los siguientes:

- **Oferta de servicios de cómputo**.

Los servicios de cómputo son un conjunto de características de hardware virtual, como, por ejemplo, la frecuencia de la CPU, el número de núcleos de procesamiento o la cantidad de memoria. Es posible crear varios servicios de cómputo para que se cumplan los requisitos constituidos. Las especificaciones de un servicio de cómputo establecen:

  - CPU
  - Memoria
  - Tasa de transferencia de red
  - Etiquetes de almacenamiento y huésped. Estas etiquetas se establecen para llevar una organización del entorno.

Se definen cinco servicios de cómputo, permitiendo al usuario final disponer de aquél que se ajuste mejor a sus necesidades. Los servicios ofertados son:

1. Instancia Diminuta o *Tiny Instance*. Esta plantilla ofrece una CPU de 1000 MHz de frecuencia y 512 MB de memoria RAM.
2. Instancia Pequeña o *Small Instance*. Esta plantilla ofrece una CPU de 1000 MHz de frecuencia y 768 MB de memoria RAM.
3. Instancia Mediana o *Medium Instance*. Se establece una plantilla con 2 CPUs de 1000 MHz y 1024 MB de memoria RAM.
4. Instancia Grande o *Large Instance*. Se establece una plantilla con 2 CPUs de 1000 MHz y 2048 MB de memoria RAM.
5. Instancia Personalizada o *Experimental Instance*. Esta plantilla es de uso exclusivo por el administrador el cual dispondrá de los recursos que se estimen en cada momento.

- **Oferta de servicios de disco.**

El servicio de oferta de disco establece las propiedades de los discos disponibles para la instalación de una máquina virtual, o como disco adicional de datos. Es posible establecer un tamaño específico o permitir al usuario elegir el tamaño del mismo. Las especificaciones de este servicio son:

  - Tamaño del disco
  - Personalización del disco
  - Etiquetas de almacenamiento para una mejor organización.

En este contexto, se definen cuatro tipos de discos para permitir al usuario final disponer del que se ajuste mejor a sus necesidades. Los discos ofertados son:

1. Disco Pequeño o *Small Disk* con un disco de 5 GB.
2. Disco Mediano o *Medium Disk* con un disco de 20 GB.
3. Disco Personalizado o *Custom Disk* con un disco de tamaño configurable por el usuario. El tamaño nunca podrá sobrepasar el máximo tamaño definido en los parámetros globales.

3.3.2.5. Configuraciones de red

Se configura el laboratorio web para poder disponer de diferentes tipos de redes. Existen dos principales redes de huéspedes. Una red compartida por todos los usuarios y proyectos con acceso a Internet, ***net_internet***, y una segunda red independiente de la primera y también compartida por todos los usuarios y proyectos pero sin salida a Internet denominada ***net_cosec.***

Los usuarios no tienen la capacidad de crear nuevas redes, ni públicas ni privadas, correspondiendo al administrador la posibilidad de crear nuevas redes virtuales, ya sean privadas para cuentas o proyectos o públicas y compartidas como las redes anteriores.

3.3.2.6. Configuración de cuentas

Se diseñan dos tipos de cuentas de usuario. Una con permisos de administrador, y una segunda con permisos de usuario. El primer tipo de cuenta, es capaz de administrar globalmente el entorno web, pudiendo modificar configuraciones a nivel general, a nivel de cuentas y a nivel de proyectos. El segundo tipo de cuenta, de usuario, tiene la capacidad de crear sus máquinas virtuales, administrar sus plantillas, ISOs…, siendo también capaz de administrar su cuenta y sus proyectos, pero sin salirse de su entorno. Cualquier configuración adicional a las básicas ofrecidas deberá ser solicitada al administrador del sistema.

3.3.3. Diagramas de Estado

Por último, en esta sección se ilustra el diseño final del laboratorio web desde diferentes puntos de vista. En este sentido, la *Figura 44. Diagrama de cómputo y almacenamiento del laboratorio web*. muestra el esquema de cómputo y almacenamiento diseñado. En él se puede apreciar la existencia de una zona y de los diferentes elementos que la componen y que se han descrito anteriormente la sección 2.5.1.

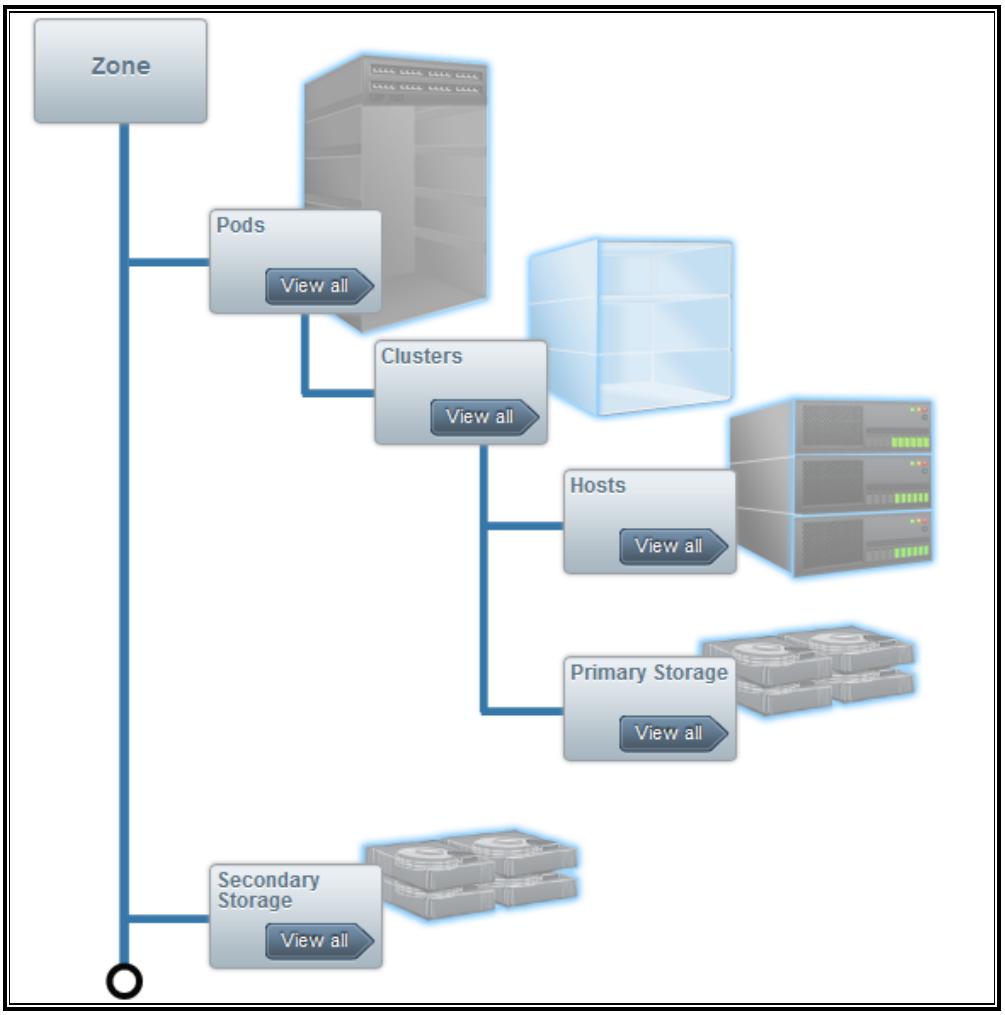


Figura 44. Diagrama de cómputo y almacenamiento del laboratorio web.[34]

El diagrama de red muestra los diferentes tipos de tráfico existentes y cómo se agrupan en una única tarjeta de red.

[34] Laboratorio web COSEC. https://www1.seg.inf.uc3m.es/

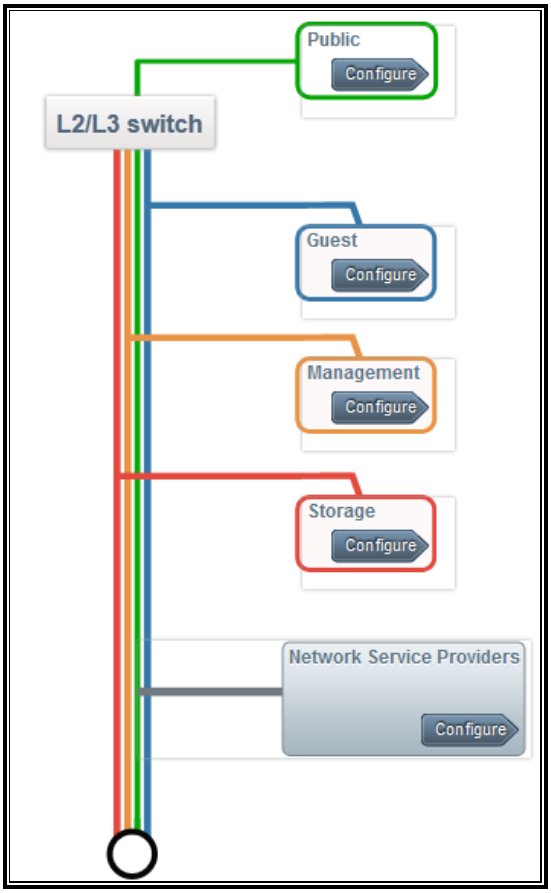


Figura 45. Diagrama de red del laboratorio web.[35]

Una de las formas de completar los diagramas y comprender mejor el diseño interno del laboratorio es el diagrama de estados de las máquinas virtuales. Las máquinas virtuales tienen 4 estados: creadas, destruidas, en ejecución o caídas. Pueden pasar de un estado a otro en función del estado que requiera el usuario de la misma. A continuación se pueden apreciar los diferentes estados así como las acciones que modifican cada uno de ellos.

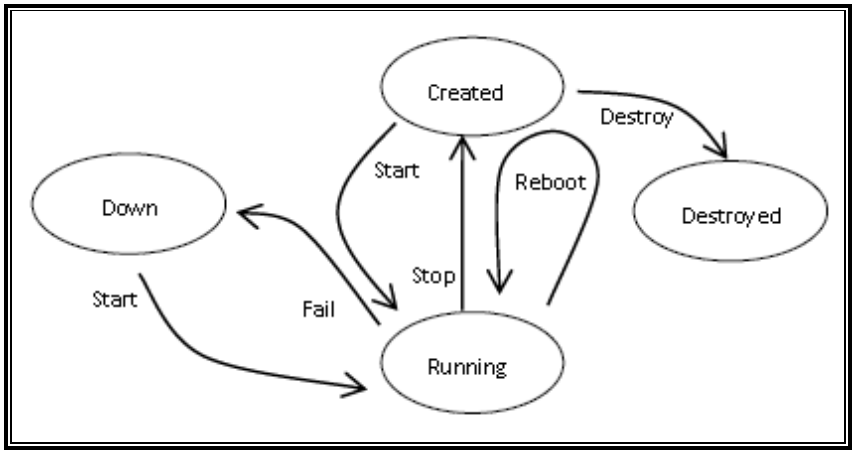


Figura 46. Diagrama de estados de una máquina virtual en el laboratorio web.

[35] Laboratorio web COSEC. https://www1.seg.inf.uc3m.es/

Capítulo 4

4. IMPLEMENTACIÓN

En el capítulo 4 se desarrolla, de forma detallada, la implementación del software en los dos laboratorios para poder obtener una visión de conjunto de las diferentes fases comentadas en capítulos anteriores.

4.1. Laboratorio automático

4.1.1. Consola gestión

Para la implementación del servidor que realiza las funciones de consola de gestión, se comienza con la instalación de una máquina con el sistema operativo Ubuntu 12.04.1 LTS. Posteriormente, una vez finalizada la instalación del sistema operativo, se procede a realizar las siguientes acciones sobre la máquina:

- En primer lugar, efectuar una actualización del sistema operativo con el comando **apt-update** seguido de un **apt-upgrade**.
- Posteriormente se continúa con la instalación de los paquetes *stunnel4* y *xcp-xe* con el comando **apt-get install stunnel4 xcp-xe**. Estos paquetes son necesarios para la ejecución remota de comandos *xe* en el hipervisor.
- El siguiente paso consiste en crear un usuario denominado **uc3m**. Este usuario básico, se utiliza para la conexión vía *ssh* desde los equipos Windows de cada experimento a la consola.
- En cuarto lugar, se procede a la instalación del paquete *denyhosts* con la ejecución del comando **apt-get install denyhosts**. Este programa, entre otras cosas, bloquea las IPs que introduzcan de forma errónea la contraseña del usuario *root* más de N veces.
- Consecutivamente, como medida adicional de seguridad, se deshabilita la conexión vía *ssh* con el usuario *root*. Para ello, es necesario editar el fichero ***ssh_config*** (**vi /etc/ssh/sshd_config**) y modificar la línea que indica ***PermitRootLogin***. Esta línea debe quedar de la siguiente forma:

PermitRootLogin no. Posteriormente hay que reiniciar el servicio de ***sshd*** (**restart ssh**) para aplicar los cambios.

Por último, se realizan las siguientes acciones:

- Copia de los programas necesarios para la configuración del laboratorio *malware* en */home*.
- Copia del repositorio de aplicaciones. Se copia el conjunto de aplicaciones establecidas a la ruta */home/apps* del servidor.
- Copia del repositorio de malware. Se copia el malware necesario a la ruta */home/malware*. En caso de que el espacio en disco sea insuficiente podría ser preciso añadir un nuevo disco a la máquina virtual. Una vez agregado el nuevo disco al equipo, es necesario realizar los siguientes pasos para que se configure correctamente:
  fdisk /dev/xvdb → Opción w para crear la tabla del disco
  mkfs.ext4 /dev/xvdb → Necesario para dar formato al disco
  partprobe /dev/xvdb → Necesario para actualizar el kernel
  mkdir /mnt/xvdb → Carpeta donde se exporta el disco
  chmod 777 /mnt/xvdb → Se dan los permisos oportunos a la carpeta creada
  mount /dev/xvdb /mnt/xvdb -t ext4 → Punto de montaje del nuevo disco en formato ext4
  Para que el cambio sea persistente es necesario modificar el fichero */etc/fstab* y añadir al final del mismo la siguiente línea: ***/dev/xvdb /mnt/xvdb ext4 defaults 0 0***

4.1.2. Máquina DHCP

En este apartado se comenta el procedimiento a seguir para la implementación de la máquina virtual encargada de desempeñar las funciones de DHCP. Como ya se ha comentado, es la primera máquina en iniciarse para proporcionar una definición de red y un rango de IPs a los clientes. Las especificaciones hardware de este servidor se han definido como las siguientes:

- 1 VCPUs
- 2048 GB de RAM
- 1 tarjeta de Red
- 20 GB de disco duro

Tras la creación de la máquina virtual, se debe realizar la instalación del sistema operativo Windows Server 2008 R2 x64 desde el DVD virtual (*es_windows_server_2008_r2_with_sp1_x64_dvd_617398.iso*). El procedimiento de instalación del sistema operativo será el estándar, seleccionando las opciones por defecto.

A continuación, tras la instalación del sistema operativo, hay que configurar el inicio de sesión automático. Los pasos a seguir están indicados de forma clara y sencilla en un artículo de Microsoft[36]. Posteriormente, se instalan los parches de sistema operativo disponibles a través de las actualizaciones automáticas de sistema. Una vez instalados, se configura el servicio de DHCP. Para configurar el servicio hay que realizar los siguientes pasos:

- Configurar una IP estática en el servidor. La IP configurada será la 10.10.10.1
- Botón derecho sobre Equipo y seleccionar Administrar.
- Una vez en el panel de administrador del servidor hay que ir a la sección de Roles y presionar sobre *Agregar Roles*. Aparece un asistente en el que hay que seleccionar el rol de *Servidor DHCP*.
- Se configura el ámbito como se muestra a continuación:

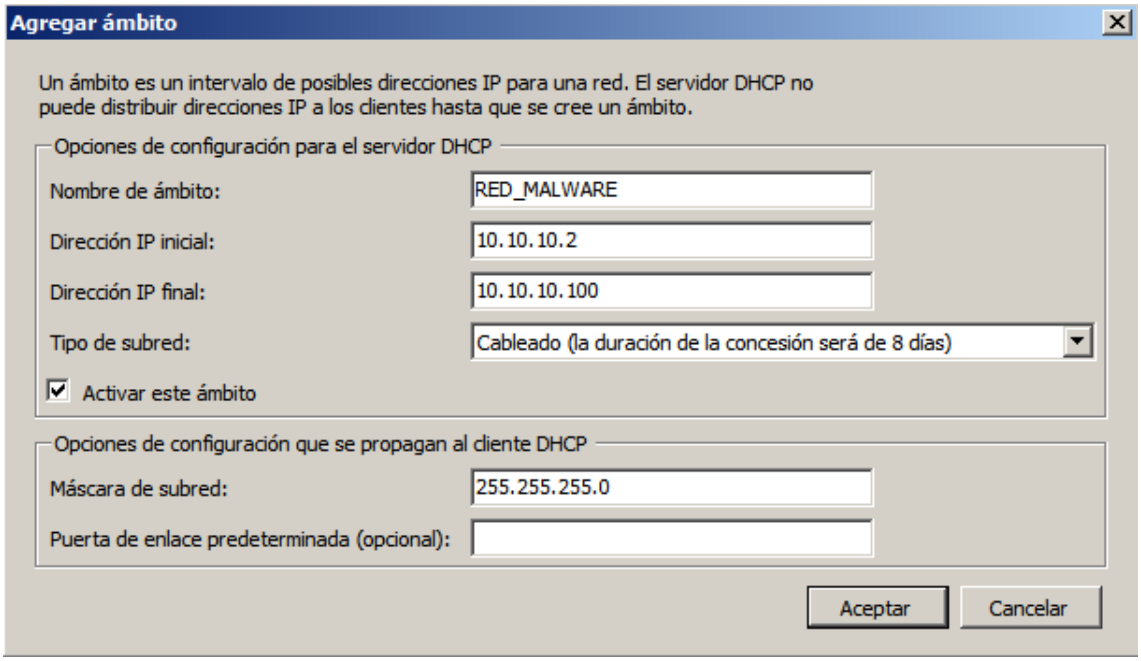


Figura 47. Configuración ámbito DHCP laboratorio automático.

- Una vez configurado el servidor DHCP, no se debe olvidar comprobar que el servidor esté funcionando correctamente. Para ello, es suficiente retornar al panel de administrador del servidor y comprobar que el servicio está activo, como se muestra en la *Figura 48. Confirmación del correcto funcionamiento del servicio de DHCP*. o, alternativamente, arrancar un equipo en esa red y comprobar que obtiene una IP correctamente.

[36] http://support.microsoft.com/kb/315231

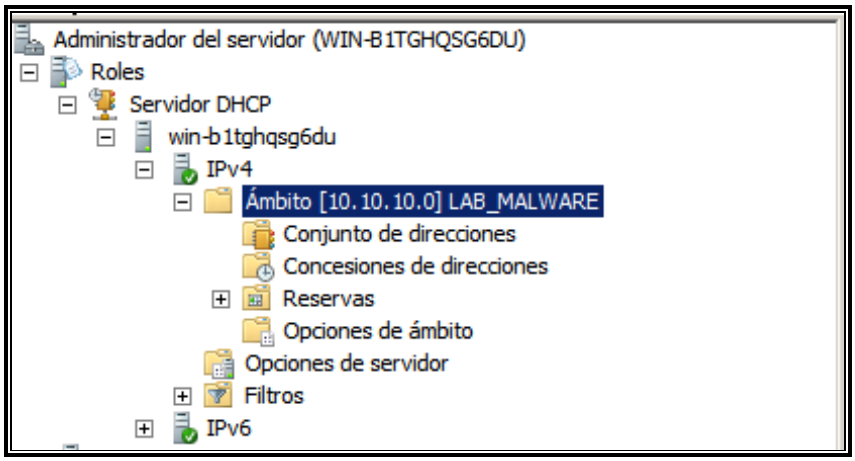


Figura 48. Confirmación del correcto funcionamiento del servicio de DHCP.

4.1.3. OSSIM

Para la implantación del sistema SIEM, se considera conveniente utilizar la versión 4.1 de OSSIM, que se puede descargar de forma gratuita de la web de *AlienVault*.[37]

A continuación, una vez descargada la imagen del sistema en el repositorio de imágenes disponible en el Abastor, se procede a crear una máquina virtual, utilizando la plantilla *Other Install Media*, en el hipervisor con, al menos, las siguientes características:

- 2 VCPUs
- 4096 GB de RAM
- 1 tarjeta de Red
- 20 GB de disco duro

Una vez creada la máquina virtual, ésta arranca desde el DVD virtual, e instala el sistema operativo. Los pasos que se han seguido para la instalación del servidor OSSIM son los que especifica el documento de instalación del fabricante.[38]

Instalado el sistema operativo, el siguiente paso consiste en configurar adecuadamente la herramienta para que las especificaciones sean las adecuadas. Esta configuración hay que realizarla iniciando sesión en el sistema con el usuario *root*, previamente configurado en instalación, en la que se ejecutan los comandos que se enuncian a continuación, y que se han obtenido del manual anteriormente mencionado. En cualquier caso, la enumeración contiene la serie de comandos ejecutados, tanto de los que se recogen en el manual, como fuera de él:

- **vi /etc/inittab** →Es necesario *comentar* la línea 54 para que no aparezca por pantalla cada 5 minutos el mensaje**:** *INIT: id "co" respawning too fast: disabled for 5 minutes***.**

[37] http://communities.alienvault.com/community/
[38] http://communities.alienvault.com/docs/Installation_Guide.pdf

- Instalación de las herramientas de *Xen*. Se hace necesario para ello cargar la ISO xs-tools.iso en el DVD virtual de la máquina y ejecutar los siguientes pasos:
    - **mkdir /mnt/xcp**
    - **mount /dev/scd0 /mnt/xcp**
    - **cd /mnt/xcp/Linux**
    - **./install.sh**

Una vez finalizado el procedimiento, es preciso reiniciar el servidor.

- **date MMDDhhmm[CC]YY[.ss]** → Este proceso sirve para corregir la fecha y hora que se establece en instalación.
- **ossim-setup** → En este caso, su ejecución tiene como finalidad modificar la configuración de los sensores y los monitores:
    - Se selecciona la opción 3 (*Change Sensor Settings*).

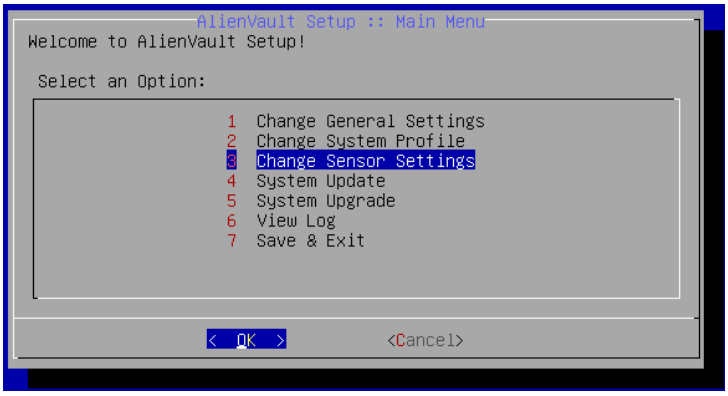


Figura 49. Pantalla Inicial al ejecutar ossim-setup en la línea de comandos.

    - Se selecciona la opción 3 (*Enable/Disable detector plugins*). Una vez seleccionada, hay que habilitar los siguientes complementos: **arpwatch, dhcp, ossec-idm, ossec-single-line, ossec-unique-line, ossec, ossim-agent, p0f, pads** entre otros**.** Una vez finalizado se confirma presionando en *OK*.

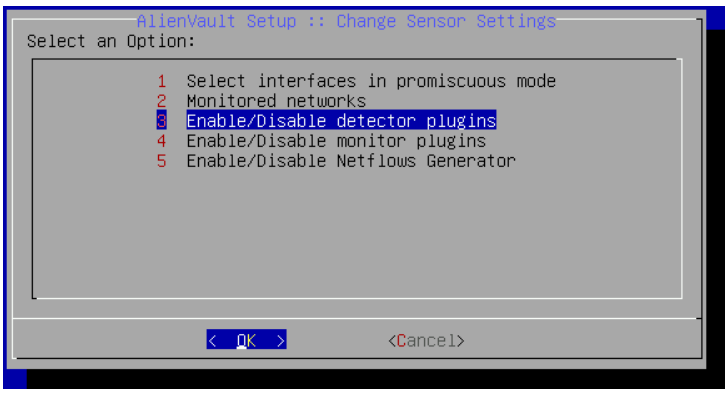


Figura 50. Pantalla con la opción resaltada de modificación de sensores.

- Posteriormente se vuelve a seleccionar la opción 3 de la pantalla inicial (*Change Sensor Settings*) y la opción 4 en la segunda pantalla (*Enable/disable monitor plugins*) para modificar los monitores correspondientes:

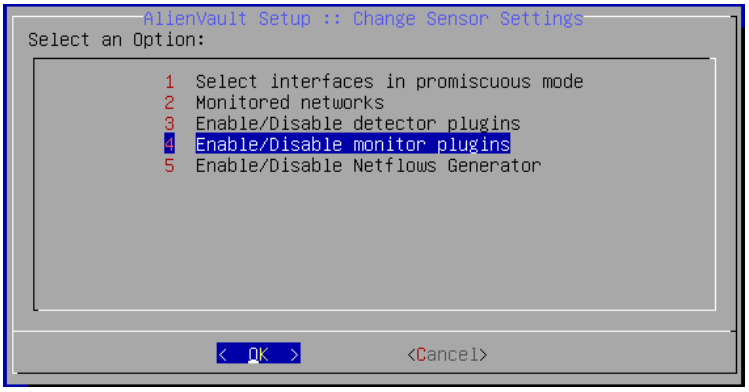


Figura 51. Figura con la opción de modificación de monitores resaltada.

Los monitores que se habilitan son: **nmap-monitor, ntop-monitor, ossim-monitor, ping-monitor, whois-monitor y wmi-monitor**.

Tras aplicar estos cambios, es imprescindible reiniciar el servicio del agente de ossim con el siguiente comando: **/etc/init.d/ossim-agent restart**.

- Por defecto, la alerta de la existencia de nuevos ficheros en los sistemas está deshabilitada. Sin embargo para la detección de malware, esta alerta es muy útil por lo que es necesario configurarla. Para su configuración, hay que incluir la siguiente regla en el fichero **vi /var/ossec/rules/local_rules.xml**[39]:
  <rule id="554" level 10 overwrite="yes">
  <category>ossec</category>
  <decoded_as>syscheck_new_entry</decoded_as>
  <description>File added to the system.</description>
  <group>syscheck,</group>
  </rule>
- Por último, se configuran los agentes oportunos para que se monitoricen. En este caso se crean 15 agentes a través del comando **/var/ossec/bin/manage-agents**. La siguiente imagen muestra los agentes configurados:

[39] http://www.ossec.net/doc/manual/syscheck/

```
Available agents:
   ID: 001, Name: alienvault, IP: 10.10.10.102
   ID: 002, Name: w002, IP: 10.10.10.2
   ID: 003, Name: w003, IP: 10.10.10.3
   ID: 004, Name: w004, IP: 10.10.10.4
   ID: 005, Name: w005, IP: 10.10.10.5
   ID: 006, Name: w006, IP: 10.10.10.6
   ID: 007, Name: w007, IP: 10.10.10.7
   ID: 008, Name: w008, IP: 10.10.10.8
   ID: 009, Name: w009, IP: 10.10.10.9
   ID: 010, Name: w010, IP: 10.10.10.10
   ID: 011, Name: w011, IP: 10.10.10.11
   ID: 012, Name: w012, IP: 10.10.10.12
   ID: 013, Name: w013, IP: 10.10.10.13
   ID: 014, Name: w014, IP: 10.10.10.14
   ID: 015, Name: w015, IP: 10.10.10.15
```

Figura 52. Agentes configurados en OSSIM.

Las claves generadas en este punto son las que hay que almacenar para crear los ficheros de configuración para cada agente. Los ficheros de configuración se almacenan en la carpeta */home/apps/ossim* de la consola. En caso de ser necesario añadir más agentes, habrá que crear los nuevos ficheros correspondientes en la consola.

El siguiente paso, tras la instalación y configuración del servidor OSSIM, consiste en efectuar la configuración del acceso web al servidor. Para ello, se procede a introducir en un navegador web la IP que se ha dispuesto en el servidor en el momento de instalación. Una vez introducida la IP en un navegador, aparece un mensaje de seguridad, ya que la conexión se realiza de forma segura a través de *https*. Es entonces cuando hay que cumplimentar el siguiente formulario, con el fin de poder empezar a utilizar la monitorización web del entorno:

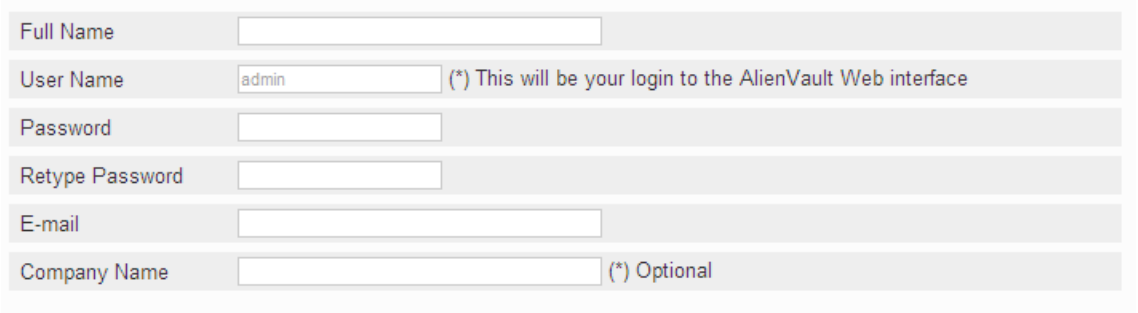


Figura 53. Configuración inicial de OSSIM.

4.1.4. Máquinas virtuales víctimas y atacantes.

Una vez implantadas las máquinas virtuales que actúan como infraestructura del laboratorio, se definen las máquinas virtuales que van a constituir el entorno de ejecución del malware. En este sentido, es posible crear máquinas virtuales bajo demanda en base a unas plantillas configuradas para que cumplan los requisitos necesarios para el entorno de laboratorio que se desea generar.

Estas plantillas, a su vez, se generan a partir de la instalación de un sistema operativo base más las configuraciones necesarias para que su procedimiento sea el esperado. Para el desarrollo de este proyecto, la operativa a realizar para generar cada plantilla será la siguiente:

- Instalación del sistema operativo base (WXP, W7, W2K3…)
- Instalación del programa gratuito WinSCP, que posibilita realizar conexiones remotas con la consola de gestión donde se encuentra el repositorio de aplicaciones y los ficheros de configuración. Para su instalación, es preciso efectuar la descarga de Internet de la aplicación y realizar la instalación por defecto. La versión instalada en las máquinas es la 5.0.8 RC.
- Configuración del sistema operativo para que inicie sesión de forma automática al iniciarse la máquina virtual
- Copiar del fichero *startup.cmd* a la carpeta de inicio del usuario con el que se realiza el inicio de sesión automática en el equipo. De esta manera, se ejecuta en el arranque de la máquina. También es necesario copiar el fichero *copia.vbs* en el directorio raíz del disco C (C:\). El funcionamiento de estos ficheros se comenta con mayor detalle más adelante.
- Instalación de las herramientas de gestión de *Xen* para Windows. Estas herramientas sirven para aumentar la funcionalidad y facilitar la interacción de los usuarios con las máquinas virtuales. Para su instalación, será necesario cargar la ISO *xs-tools.iso* en el DVD virtual de la máquina. Una vez cargado, se ejecuta el fichero *xensetup.exe* siguiendo los pasos por defecto que muestra el instalador. Tras la instalación, se procede de nuevo a reiniciar el equipo para que se apliquen los cambios.
- Deshabilitar el Firewall y el control de acceso de usuario, UAC, si procede.
- Copia de la aplicaciones a C:\apps para evitar la copia de todas las aplicaciones en el inicio de la máquina y agilizar la ejecución del laboratorio.
- Instalación del programa que permite capturar los eventos que se generen en el equipo Windows para, posteriormente, enviarlos al servidor OSSIM. El programa que se instala se denomina agente *OSSEC* para Windows. Para su instalación y configuración, se deben seguir los pasos que se indican en el manual del producto.[40]

  En primer lugar, es necesario realizar la descargar del ejecutable de la siguiente url: http://www.ossec.net/?page_id=19. Una vez descargado, hay que instalar el ejecutable con la configuración por defecto. Posteriormente, cuando se configuren los diferentes experimentos, se copian los correspondientes ficheros de configuración para cada sistema para conseguir un correcto funcionamiento

[40] http://www.ossec.net/?page_id=11

del agente. A modo de resumen, existen 3 ficheros importantes que se copian a la hora de la configuración:

- **client.keys**: En este fichero se almacenarán las diferentes claves privadas que se utilizan para conexión con OSSIM. Las claves se obtienen de ejecutar el comando **/var/ossec/bin/manage-agents** comentado anteriormente.
- **internal_options**: Donde se configuran opciones de rendimiento del agente.
- **ossec.conf**: Se trata del fichero de configuración del agente. En él se indican los ficheros, las carpetas, las claves de registro…que se van a monitorizar, así como los elementos que se deben excluir de la monitorización.

Una vez copiados los ficheros, hay que reiniciar el servicio *OSSEC HIDS (**OssecSvc**)* para que los cambios sean efectivos.

Por último, finalizada la instalación y configuración de la máquina virtual base correspondiente a cada sistema operativo, se procede a realizar su conversión a plantilla para que posteriormente pueda ser utilizada como patrón para la generación de nuevas máquinas virtuales. Para llevar a cabo esta operativa, en la consola de administración de *Xen* (XenCenter), se selecciona la máquina virtual correspondiente, que tiene que estar apagada, hacer clic con el botón derecho del ratón y escoger la opción de convertir a plantilla (*Convert To Template*). Veamos gráficamente, en las figuras las correspondientes, las acciones que se deben llevar a cabo:

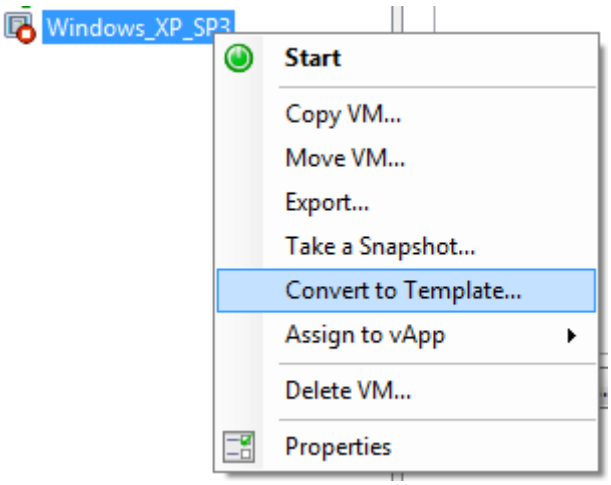


Figura 54. Conversión a plantilla de una máquina virtual.

Hay que confirmar que se desea convertir la máquina en plantilla para finalizar con la operación.

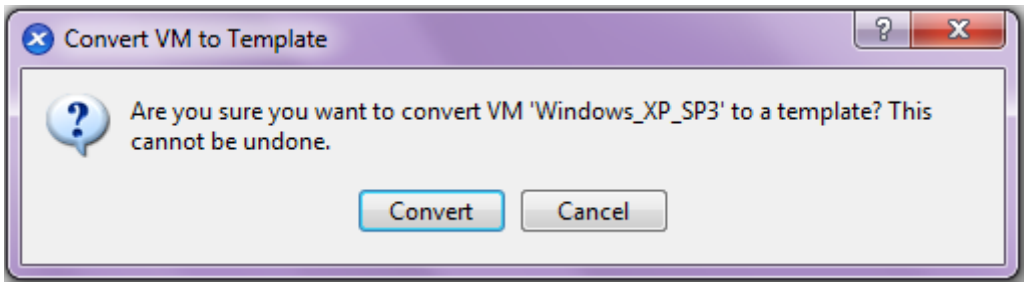


Figura 55. Confirmación de la conversión a plantilla de una máquina virtual.

Tras esta operativa, se genera una plantilla a partir de la que se podrán iniciar nuevas máquinas virtuales. Para ello, es necesario realizar los pasos descritos anteriormente en cada una de las plantillas que se deseen utilizar para el laboratorio.

4.1.5. Generación de Scripts

Para automatizar en la medida de lo posible la ejecución del laboratorio malware, es necesaria la generación de varios scripts. El primero de ellos, es el encargado de generar la infraestructura del laboratorio y se ejecuta en la consola de gestión. El resto, se ejecutan en cada máquina virtual realizando lo pasos oportunos para su correcta configuración.

4.1.5.1. Script de configuración del laboratorio.

Se trata de un script cuya realización se efectuó en lenguaje propio de la *Shell* de Linux. Su ejecución se realiza en la consola de gestión. Debido a las medidas de seguridad adoptadas, hay que realizar la conexión a la consola de gestión mediante *ssh* desde la red de la Universidad. En la ruta donde esté alojado el programa, se deberá ejecutar el siguiente comando:

```
# ./lab_automatico.sh
```

Este programa ejecuta todas las operaciones necesarias para configurar el experimento que se ha generado. El código del mismo se adjunta en la documentación del proyecto. A continuación se describen únicamente los parámetros de entrada del programa:

- **-h** Muestra la ayuda que ofrece el programa con las diferentes opciones disponibles.
- **-n [numero]** Representa el número de máquinas de cada tipo que se van a crear en el experimento. El máximo número de máquinas de cada tipo que se pueden crear es de 3. Este parámetro no es obligatorio. Si no se establece este parámetro, se crea un número aleatorio de máquinas virtuales.
- **-t [numero][s|m|h|d]** Indica el tiempo en segundos, minutos, horas o días durante el que se ejecuta el experimento. Este parámetro es imperativo.
- **-e [nombre]** Indica el nombre del malware que se ejecuta dentro del experimento. Este parámetro también será obligatorio. En caso de no indicarse, el experimento se lanza sin malware.

4.1.5.2. Programas de configuración de las máquinas virtuales.

Como ya se ha visto anteriormente, una vez se iniciadas las máquinas virtuales, el programa principal se mantiene a la espera de que finalicen de configurarse las máquinas virtuales. Estas máquinas virtuales tienen que ejecutar una serie de programas para cumplir con las especificaciones establecidas. Son tres los programas relevantes que se ejecutan en las máquinas virtuales.

- El primero de ellos, **copia.vbs**, es un programa previamente situado en el raíz de la máquina virtual, que sincroniza las aplicaciones disponibles en la consola. Una vez sincronizadas, copia los ficheros **global.conf, startup.vbs** e **instalación.vbs** a *C:\apps*. Además, deshabilita el servicio de OSSEC para evitar que se almacenen eventos irrelevantes en el experimento mientras éste se configura. Por último, ejecuta el programa **startup.vbs**.
- El segundo programa, **startup.vbs**, obtiene del fichero **global.conf** el identificador del experimento y copia el perfil concreto que le corresponde a esa máquina virtual en base al experimento que se esté llevando a cabo. También configura el equipo para que, después del reinicio, ejecute el programa **inatalacion.vbs**. Por último, modifica el nombre de la máquina para que no existan nombres duplicados en la red y reinicia el equipo.
- El tercer programa, **instalación.vbs**, se ejecuta después del reinicio anterior de la máquina virtual e instala las aplicaciones que vengan determinadas en el perfil de la máquina para, una vez instaladas, enviar una bandera a la consola para indicar que ha finalizado de configurarse. Se especifica un bucle que espera a recibir la señal de la consola que le indica que puede proseguir con su ejecución. Esto se produce cuando se recibe el fichero **fin.txt** por parte de la consola. Tras recibir el fichero, se configura el equipo para que ejecute el *malware* después de reiniciar la máquina virtual, en caso de que así se establezca en el perfil del equipo. También copia los ficheros de configuración de ossec comentados anteriormente (**client.keys, internal_options y ossec.conf**) y habilita el servicio de ossec para que, después de reiniciar, el agente comience a recoger datos. En último lugar, reinicia la máquina.

4.2. Laboratorio Web

Para la implementación del laboratorio web se ha utilizado CloudStack en base a las razones expuestas en el apartado ***2.5.3 Comparativa OpenStack vs CloudStack***. A continuación se desarrollan las implementaciones de los diferentes elementos y configuraciones.

4.2.1. WebFrontEnd

La máquina WebFrontEnd corresponde con la máquina virtual donde se realiza la instalación del software de CloudStack y de las bases de datos. Asimismo, es la pasarela de acceso web de los usuarios.

Para la instalación y configuración de la consola web de CloudStack o WebFrontEnd, se ha utilizado el modelo comentado en el apartado de diseño, donde la consola web es una máquina virtual más en el hipervisor; en este caso en otro hipervisor independiente. Para la instalación de esta consola, se instala, en primer lugar, un servidor virtual con un Ubuntu Server 10.04 LTS de 64 bits. Una vez instalado el sistema operativo y actualizado con los últimos parches del sistema operativo, se realizan las configuraciones oportunas para instalar y desarrollar el entorno. La descripción de los pasos seguidos se encuentra en el *ANEXO I: Instalación y Configuración de CloudStack.*

El siguiente paso, tras la instalación y configuración de CloudStack, es tomar las medidas oportunas para que la máquina de acceso web esté blindada ante posibles agujeros de seguridad. Para ello, se especifican unas reglas en el cortafuegos que únicamente permiten el acceso al equipo a través de los puertos tcp 80 y 443 desde la direcciones IP de la Universidad, 163.117.0.0/16. Fuera de este rango de direcciones, no se puede realizar ninguna conexión al servidor. A continuación se muestran las reglas utilizadas en el fichero *iptables_web.sh* del servidor WebFrontEnd:

```
#!/bin/bash

# Delete any previous settings
iptables -F
iptables -F INPUT
iptables -F OUTPUT
iptables -F FORWARD

# Default policy
iptables -P INPUT DROP

# Allow all inbound traffic from loopback interface
iptables -A INPUT -i lo -j ACCEPT

# Allow inbound TCP at port 80 and 443 from remote address
iptables -A INPUT -i eth0 -p tcp --dport 80 -s 163.117.0.0/16 -j ACCEPT
iptables -A INPUT -i eth0 -p tcp --dport 443 -s 163.117.0.0/16 -j ACCEPT
```

```
# Allow already stablished connections
iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
iptables -A OUTPUT -m state --state ESTABLISHED,RELATED -j ACCEPT

# log pings (ICMP type 8)
iptables -A INPUT -p icmp --icmp-type 8 -j LOG --log-prefix "Ping pkt received::
"
# log ssh attempts
iptables -A INPUT -p tcp --dport 22 -j LOG --log-prefix "Unauth SSH conn::"

iptables-save
```

Adicionalmente, para evitar fisuras de seguridad, el fichero se debe ejecutar cada vez que se reinicia el servidor para así, impedir que la máquina quede expuesta a posibles ataques. Los pasos a realizar para configurar este comportamiento, son los siguientes:

- Proporcionar los permisos de ejecución oportunos **sudo chmod 777 iptables_web.sh**
- Crear un link simbólico al fichero donde se encuentran las reglas desde la carpeta ***/etc/init.d***:
  sudo ln -s /home/cosec/iptables_web.sh /etc/init.d/iptables_web.sh
- Actualizar la tabla de los enlaces simbólicos para que tenga en cuenta este nuevo fichero:
  sudo update-rc.d iptables_web.sh defaults

Una vez completados estos pasos, la máquina ejecutará el fichero en cada reinicio de la misma, estando protegida de esta forma en todo momento.

4.2.2. SSHFrontEnd

La máquina virtual, denominada SSHFrontEnd, permite el acceso remoto a las máquinas virtuales a través del protocolo SSH. Esta consola de acceso al laboratorio mediante *ssh* se implanta a partir de la plantilla de Ubuntu Server generada para el laboratorio web (Ubuntu Server 12.04 LTS x64). Al tratarse de una máquina virtual ajena a la infraestructura y concepto de CloudStack, se ha considerado conveniente independizar su instalación y administración del entorno de CloudStack, realizando una instalación directamente sobre el hipervisor a través de la consola de Xen. Para ello, se siguieron una serie de pasos a fin de obtener la máquina deseada que fueron los siguientes:

- En primer lugar, se realiza la importación de la máquina virtual UbuntuServer.xva en el hipervisor a través de la consola Xen. Este archivo está disponible en el repositorio NFS_ISO del sistema de almacenamiento Abastor y se trata de un Ubuntu Server instalado y configurado.

- Una vez importada, se modifica el nombre de la máquina virtual en la consola a SSHFrontEnd y se reducen los recursos de la máquina virtual para que disponga de una CPU virtual y 512MB de memoria RAM.

- Posteriormente, una vez encendida la máquina virtual, se configura la red pública. A través de la consola, se añade la red **Network 0** en la interfaz 0 en caso de no disponer ya de ella. Se referirá a la de red pública en adelante como la eth0. A continuación, se pasa a configurar la IP pública del servidor, que en este caso corresponde con la IP 163.117.149.66. Hay que editar el fichero /etc/network/interfaces (**sudo vi /etc/network/interfaces**) y añadir o configurar las siguientes líneas para establecer los parámetros oportunos en la eth0:

  auto eth0
  iface eth0 inet static
  address 163.117.149.66
  netmask 255.255.255.0
  broadcast 163.117.149.255
  gateway 163.117.149.2
  dns-nameservers 163.117.131.31

  Una vez guardado el fichero con la nueva configuración se procede a reiniciar el servicio de red (**sudo /etc/init.d/networking restart**) o la máquina para que se apliquen los cambios.

- Tras configurar la tarjeta de red pública, se configura el nombre del servidor, modificando los ficheros **/etc/hostname** y **/etc/hosts** para disponer correctamente el nombre del equipo como SSHFrontEnd.

- Como medida de seguridad, únicamente se conectarán a este entorno los usuarios configurados con su correspondiente clave pública para que el acceso sea mediante intercambio de claves. Para añadir los usuarios, se ejecuta el comando **sudo adduser username**, donde *username* establece el nombre del usuario que se quiere crear. Los datos que se configuran para cada usuario son los mismos que se utilizaron en la creación de los usuarios en Cloudstack.

  Los usuarios configurados en este entorno fueron gtangil, aalcaide, jbalis, jestevez, jfuentes y spastran.
  Una vez creados los usuarios, es necesario realizar una serie de operativas con el fin de configurar las claves públicas de cada usuario, evitando así el uso de contraseñas en claro. La plantilla de Ubuntu Server tiene una serie de características de seguridad que se verán más adelante. Para obtener el resultado esperado, hay que iniciar sesión con cada uno de los usuarios y ejecutar los siguientes comandos:
  mkdir .ssh
  wget username.pub (donde username.pub indica la url donde se encuentra la clave pública de cada usuario)

cat username.pub >> .ssh/authorized_keys
rm username.pub

Adicionalmente, con el fin de permitir la conexión ssh con los servidores y escritorios Linux dentro del laboratorio web, se configura en la carpeta de cada usuario la clave privada del usuario cosec, con la que se realizarán las conexiones ssh sobre las máquinas virtuales privadas. Basta con copiar a la carpeta .ssh de cada usuario el fichero **id_rsa** para la conexión remota a los servidores de la red interna sin necesidad de utilizar contraseña. Inicialmente se realiza la operativa sobre el usuario gtangil para posteriormente replicar la configuración al resto de usuarios

sudo cp /home/gtangil/.ssh/id_rsa /home/aalcaide/.ssh/id_rsa
sudo cp /home/gtangil/.ssh/id_rsa /home/jbalis/.ssh/id_rsa
sudo cp /home/gtangil/.ssh/id_rsa /home/jestevez/.ssh/id_rsa
sudo cp /home/gtangil/.ssh/id_rsa /home/jfuentes/.ssh/id_rsa
sudo cp /home/gtangil/.ssh/id_rsa /home/spastran/.ssh/id_rsa

Y establecer los permisos oportunos:

chown aalcaide:aalcaide /home/aalcaide/.ssh/id_rsa
chown jbalis:jbalis /home/jbalis/.ssh/id_rsa
chown jestevez:jestevez /home/jestevez/.ssh/id_rsa
chown jfuentes:jfuentes /home/jfuentes/.ssh/id_rsa
chown spastran:spastran /home/spastran/.ssh/id_rsa
chmod 600 id_rsa (En cada carpeta de usuario)
chmod 700 .ssh (Permisos sobre la carpeta ssh)
chmod 644 .ssh/authorized_keys (Permisos sobre el fichero de claves)

- Después, es preciso configurar el servidor para que tenga visibilidad con las diferentes redes que se vayan incorporando a CloudStack. Inicialmente se configura en el servidor las redes *net_internet* y *net_cosec*. Los pasos a seguir serían los mismos para las nuevas redes a implementar, indicando los nuevos parámetros configurados. Se añade la red oportuna a través de la consola Xen, anotando la posición de la tarjeta de red asignada. Una vez asignada, se inicia el servidor y se le asigna la IP correspondiente. En nuestro caso la IP para la red *net_internet* es la 10.2.1.1 ya que actúa como puerta de salida de la misma y para la red *net_cosec* es la 10.2.2.2 al no tener salida a Internet.

  También hay que editar el fichero /etc/network/interfaces (**sudo vi /etc/network/interfaces**) y añadir las siguientes líneas para configurar, en este caso, la eth1 y la eth2:

  auto eth1
  iface eth1 inet static
  address 10.2.1.1
  netmask 255.255.255.0
  broadcast 10.2.1.255

network 10.2.1.0

auto eth2
iface eth2 inet static
address 10.2.2.2
netmask 255.255.255.0
network 10.2.2.0
broadcast 10.2.2.255

- En tercer lugar, para evitar el uso intensivo de IPs públicas, la red con salida a Internet utiliza una serie de herramientas para que desde esa red redirija los paquetes desde la red interna a la pública. Este procedimiento se realiza a través del servidor SSHFrontEnd y se conoce como NAT[41]. Las herramientas se instalan con el comando **sudo apt-get install bridge-utils dnsmasq**. Una vez instaladas, hay que editar el fichero **/etc/sysctl.conf** (**sudo vi /etc/sysctl.conf**) para habilitar la redirección IP. Hay que eliminar el comentario de la línea que indica: **net.ipv4.ip_forward=1**.
- Por último, se configura el cortafuegos del servidor para que permita la salida del tráfico desde la red interna y cumpla con las medidas de seguridad establecidas, que consisten en que únicamente se pueda hacer ping y conexión ssh al servidor desde la red de la Universidad (163.117.0.0/16). El fichero utilizado es el siguiente:

```
#!/bin/sh

#Primero se aceptan las conexiones entrantes, luego las redirige en la tabla FORWARD y luego se le añade una regla NAT que permite enrutar hacia afuera.
EXTERNAL_INTERFACE=eth0
EXTERNAL_DIR=163.117.0.0/16
INTERNAL_INTERFACE=eth1
INTERNAL_INTERFACE_2=eth2
INTERNAL_DIR=10.2.0.0/16

# Delete all previously existing rules
iptables -F
iptables -F INPUT
iptables -F OUTPUT
iptables -F FORWARD
iptables -t nat -F
iptables -t mangle -F
iptables -X

# default policy
iptables -P INPUT DROP

# allow all inbound traffic from loopback interface
```

[41] http://en.wikipedia.org/wiki/Network_address_translation

```
echo "Allowing all inbound traffic from loopback..."
iptables -A INPUT -i lo -j ACCEPT

# Allow established connections
iptables -A INPUT -i $EXTERNAL_INTERFACE -p tcp --dport 22 -s $EXTERNAL_DIR -j ACCEPT

iptables -A INPUT -i $EXTERNAL_INTERFACE -p icmp --icmp-type 8 -s $EXTERNAL_DIR -m state
--state NEW,ESTABLISHED,RELATED  -j ACCEPT
iptables -A OUTPUT -i $EXTERNAL_INTERFACE -p icmp --icmp-type 0 -d $EXTERNAL_DIR -m
state --state ESTABLISHED,RELATED  -j ACCEPT

iptables -A INPUT -i $INTERNAL_INTERFACE -p icmp --icmp-type 8 -s $INTERNAL_DIR -m state
--state NEW,ESTABLISHED,RELATED  -j ACCEPT
iptables -A OUTPUT -i $INTERNAL_INTERFACE -p icmp --icmp-type 0 -d $INTERNAL_DIR -m
state --state ESTABLISHED,RELATED  -j ACCEPT

iptables -A INPUT -i $INTERNAL_INTERFACE_2 -p icmp --icmp-type 8 -s $INTERNAL_DIR -m
state --state NEW,ESTABLISHED,RELATED  -j ACCEPT
iptables -A OUTPUT -i $INTERNAL_INTERFACE_2 -p icmp --icmp-type 0 -d $INTERNAL_DIR -m
state --state ESTABLISHED,RELATED  -j ACCEPT

# allow already stablished connections
iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
iptables -A OUTPUT -m state --state ESTABLISHED,RELATED -j ACCEPT

iptables -A FORWARD -i $EXTERNAL_INTERFACE -o $INTERNAL_INTERFACE -m state --state
ESTABLISHED,RELATED -j ACCEPT

# Allow outgoing connections from the LAN side
iptables -A FORWARD -i $INTERNAL_INTERFACE -o $EXTERNAL_INTERFACE -j ACCEPT

# Masquerade
iptables -t nat -A POSTROUTING -o $EXTERNAL_INTERFACE -j MASQUERADE

# log pings (ICMP type 8)
iptables -A INPUT -p icmp --icmp-type 8 -j LOG --log-prefix "Ping pkt received::"

# log ssh attempts
iptables -A INPUT -p tcp --dport 22 -j LOG --log-prefix "Unauth SSH conn::"

# save everything!
echo "Saving settings..."
iptables-save
```

4.2.3. *Creación de cuentas y proyectos*

Para crear una cuenta de usuario o un proyecto, hay que acceder al menú de Cuentas o *Accounts*.

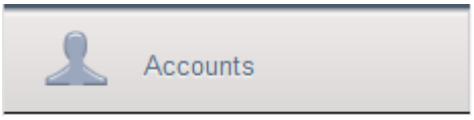


Figura 56. Gestión de cuentas de usuarios.

Para añadir una nueva cuenta, hay que ir a la sección de crear cuenta o *add account* y configurar la cuenta según las características del usuario.

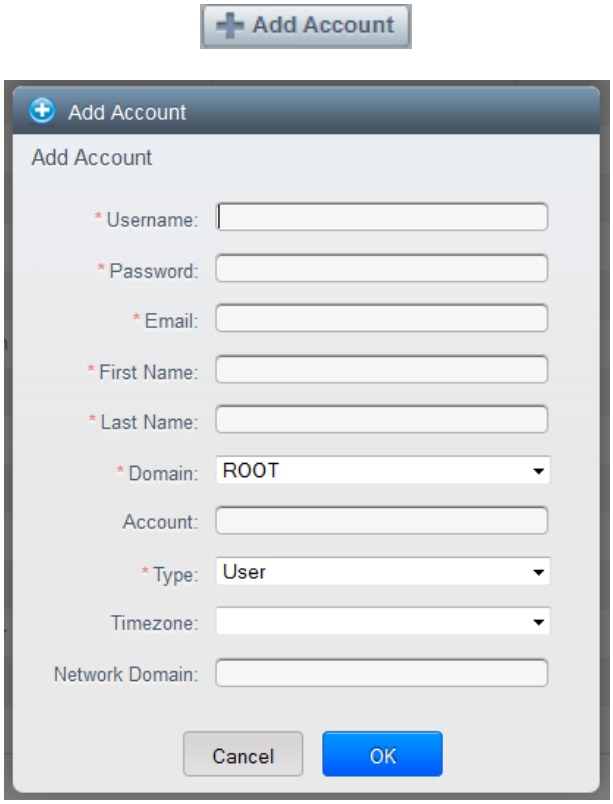


Figura 57. Creación de una cuenta de usuario.

Adicionalmente, es posible modificar los parámetros de una cuenta, eliminarla o modificar la contraseña de una de ellas. Todas estas operativas se realizan desde el menú Cuentas o *Accounts*. También es posible agregar más de un usuario a una cuenta pero, en este caso, el acceso se ha diseñado para que una cuenta equivaliese a un usuario.

Figura 58. Gestión de proyectos.

En el caso de que se desee crear un proyecto, hay que dirigirse a la sección de proyectos, donde se puede crear un nuevo proyecto o modificar uno existente. Para crear un nuevo proyecto, se utiliza el botón crear proyecto o *Create Project.*

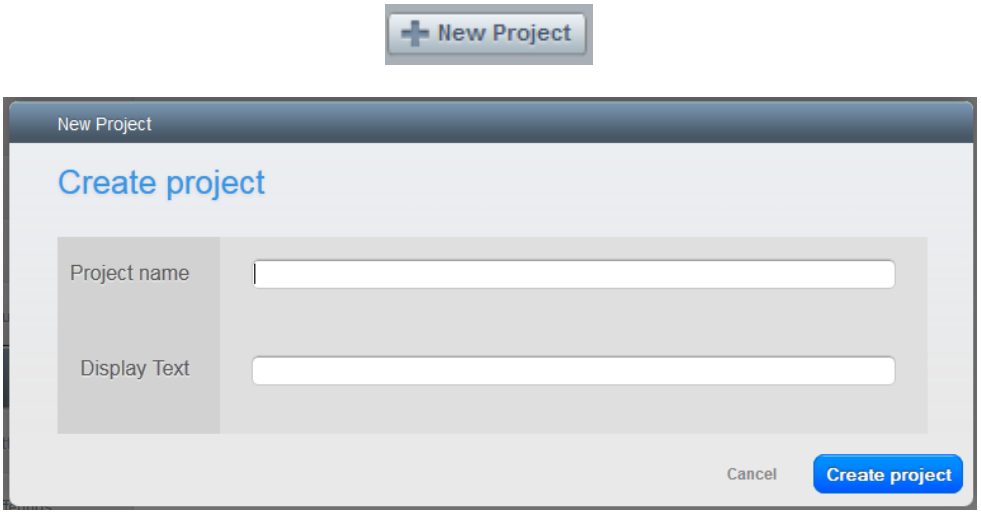


Figura 59. Creación de un nuevo proyecto.

Una vez creado, es posible seleccionar el proyecto para modificar sus detalles, añadir cuentas de usuario con visibilidad con el proyecto, o modificar los recursos asignados al proyecto.

4.2.4. Creación de redes virtuales

Uno de los requisitos existentes en la generación del laboratorio web es la creación de dos redes compartidas por todos los usuarios y proyectos. Una de las redes es interna y aislada, es decir, sin salida a Internet para la ejecución de máquinas virtuales en un entorno de ejecución particular en los que no existe necesidad de salida a Internet. La otra red es una red también interna, pero con un punto de salida al exterior. La configuración de ambas redes es idéntica a nivel de CloudStack. Posteriormente se configura un enrutador para la red con necesidad de salida a Internet, que está implementado en el servidor SSHFrontEnd como se ha comentado anteriormente. Para crear una red compartida, hay que configurar una red de huéspedes compartida. Para ello, se va a la configuración de las redes de invitados: **Infraestructure → View All → COSEC → Physical Network → Physical Network 1 → Guest (Configure) → Network → Add guest network**. Aparece una ventana donde hay que introducir la información necesaria:

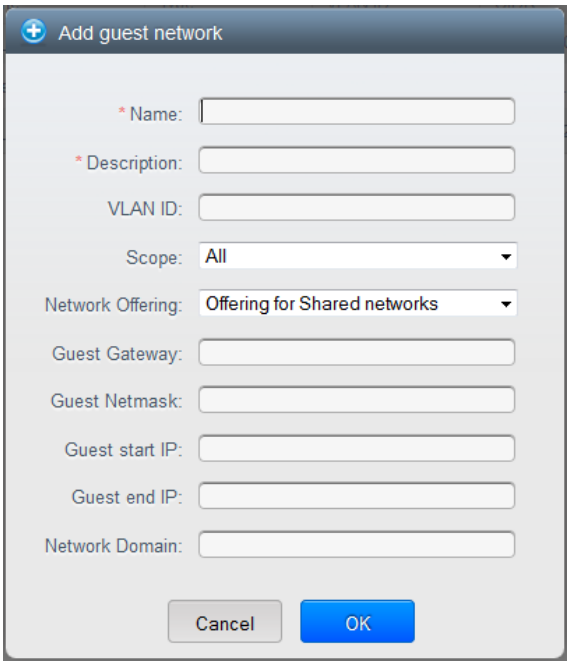


Figura 60. Configuración de redes de huéspedes.

Para el caso de la red **net_internet**, los datos que se configuran son los siguientes:

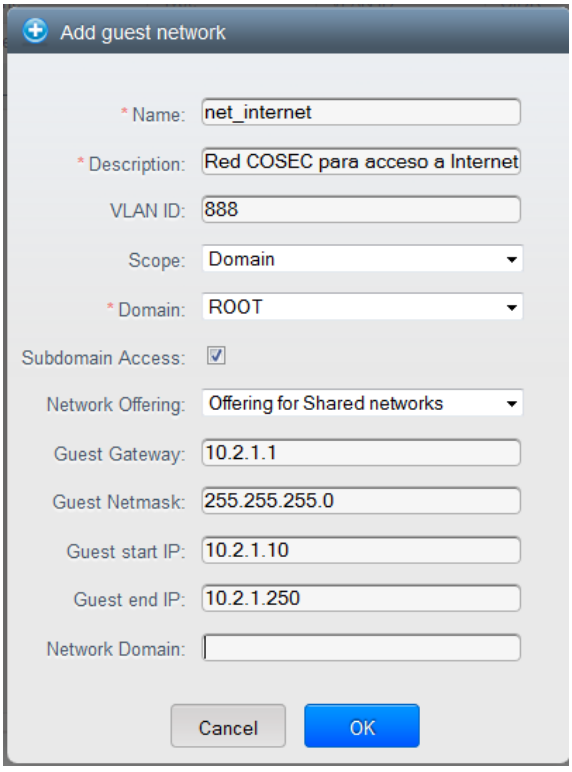


Figura 61. Configuración de la red net_internet.

En el caso de **net_cosec**, los datos son los siguientes:

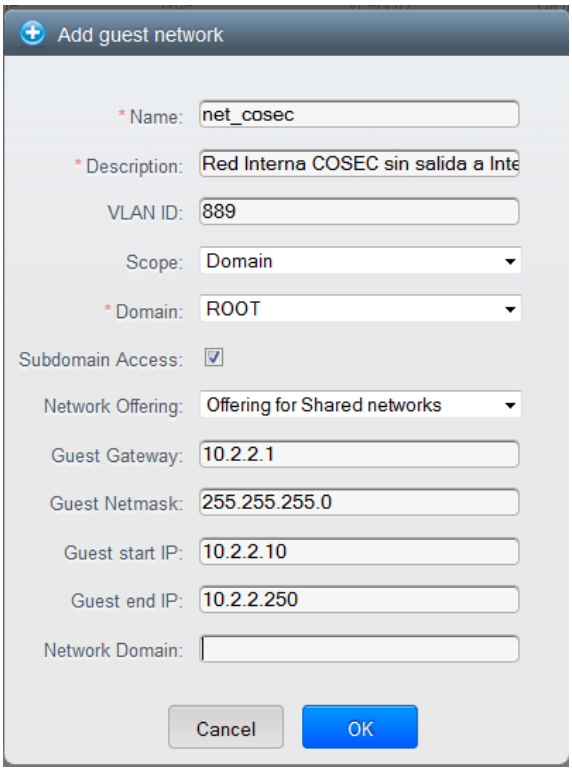


Figura 62. Configuración de la red net_cosec.

4.2.5. *Plantilla Ubuntu Server*

Para la generación de la plantilla de Ubuntu Server, se utiliza como sistema operativo un Ubuntu 12.04 LTS (Ubuntu Precise). La instalación del sistema operativo se realizada de acuerdo con el procedimiento que se describe en el *ANEXO II. Instalación de sistemas operativos Ubuntu*, donde se indican los pasos necesario para la instalación del sistema operativo. Durante la instalación, se configura al usuario ***cosec*** como único usuario de la máquina virtual con permisos de administrador. Una vez instalado, se realizan los siguientes pasos para configurar correctamente la plantilla según las necesidades establecidas.

- **sudo apt-get update**
- **sudo apt-get upgrade**

Con estos comandos se actualiza el sistema operativo con las últimas actualizaciones disponibles. También se adoptan las medidas de seguridad oportunas para evitar cualquier intrusión en la máquina. Las medidas que se llevan a cabo son las de instalar el programa *denyhosts*, la creación de un programa que configura el firewall del equipo al iniciarse el mismo y la instalación de un certificado para que únicamente se pueda realizar conexiones ssh a la máquina a través de certificado con clave pública y privada, sin posibilidad de introducir una contraseña. En primer lugar se realiza la instalación del programa *denyhosts*:

- **sudo apt-get install denyhosts**

Posteriormente, se configura el cortafuegos de la máquina para permitir, exclusivamente, el acceso a la misma desde las redes internas de Cloudstack (10.2.0.0/16) y únicamente a través de ***ssh***. El fichero que contiene las reglas correspondientes es el siguiente:

```
#!/bin/bash

INTERNAL_INTERFACE=eth0
INTERNAL_DIR=10.2.0.0/16

# Delete any previous settings
iptables -F
iptables -F INPUT
iptables -F OUTPUT
iptables -F FORWARD

# Default policy
iptables -P INPUT DROP

# Allow all inbound traffic from loopback interface
iptables -A INPUT -i lo -j ACCEPT

# Allow inbound SSH at port 22 from remote address
iptables -A INPUT -i $INTERNAL_INTERFACE -p tcp --dport 22 -s $REMOTE_DIR -j ACCEPT

# Allow inbound/outbound PING at port 22 from remote address
iptables -A INPUT -i $INTERNAL_INTERFACE -p icmp --icmp-type 8 -s $REMOTE_DIR -m
 state --state NEW,ESTABLISHED,RELATED  -j ACCEPT
iptables -A OUTPUT -i $INTERNAL_INTERFACE -p icmp --icmp-type 0 -d $REMOTE_DIR -
m state --state ESTABLISHED,RELATED  -j ACCEPT
iptables -A OUTPUT -i $INTERNAL_INTERFACE -p icmp --icmp-type 8 -d 0/0 -m state
--state NEW,ESTABLISHED,RELATED -j ACCEPT
iptables -A INPUT -i $INTERNAL_INTERFACE  -p icmp --icmp-type 0 -s 0/0  -m state
 --state ESTABLISHED,RELATED -j ACCEPT

# Allow already established connections
iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
iptables -A OUTPUT -m state --state ESTABLISHED,RELATED -j ACCEPT

# Log pings (ICMP type 8)
iptables -A INPUT -p icmp --icmp-type 8 -j LOG --log-prefix "Ping pkt received::
"

# Log ssh attempts
iptables -A INPUT -p tcp --dport 22 -j LOG --log-prefix "Unauth SSH conn::"

# Save everything!
iptables-save
```

Para evitar fisuras de seguridad, el fichero se debe ejecutar cada vez que se reinicie el servidor y así impedir que la máquina quede expuesta a posibles ataques. Los pasos a realizar para ello son:

- Proporcionar los permisos de ejecución oportunos **sudo chmod 777 iptables.sh**

- Crear un link simbólico al fichero donde se encuentran las reglas desde la carpeta ***/etc/init.d***:
  sudo ln -s /home/cosec/iptables.sh /etc/init.d/iptables.sh
- Actualizar la tabla de los enlaces simbólicos para que tenga en cuenta este nuevo fichero:
  sudo update-rc.d iptables.sh defaults

Una vez completados estos pasos, la máquina ejecutará el fichero en cada reinicio de la misma, estando protegida así en todo momento.

Por último, como medida adicional de seguridad, se decide configurar la conexión remota por ***ssh*** para que se pueda realizar solamente con certificado. Se configura el certificado público en el servidor y se configura el equipo para que sólo se permita la conexión mediante certificado. La secuencia de comandos que configuran lo comentado anteriormente es la siguiente:

- En primer lugar, se copia la clave pública (***cosec.pub***) al servidor. Se puede realizar la copia mediante WinSCP, SCP…
- Una vez se disponga de la clave pública en el servidor, se crea la carpeta .ssh en el directorio del usuario cosec y se copia el contenido de la clave pública al fichero de claves autorizadas:

  cd /home/cosec
  mkdir .ssh
  cat cosec.pub >> .ssh/authorized_keys
- Por último, se configura ***ssh*** para evitar la conexión mediante contraseña, siendo la conexión permitida únicamente mediante certificado. Para ello, se procede a modificar el fichero ***sshd_config*** (**sudo vi /etc/ssh/sshd_config*)*** configurando en el mismo los siguientes parámetros:

  PubkeyAuthentication yes

  ChallengeResponseAuthentication no

  PasswordAuthentication no

  UsePAM no

Finalmente, una vez finalizada la configuración de la plantilla de Ubuntu Server, se procede a apagar el servidor para importarlo como plantilla en CloudStack. Para importar el sistema operativo, se siguen los pasos que se indican en el *ANEXO III. Creación de plantillas e imágenes en CloudStack. Subida de ficheros vhd e iso.*, dentro de la sección *Subida de discos duros virtuales o ficheros vhd*. Los datos introducidos, en el caso de un Ubuntu server paravirtualizado, son los que se muestran en la *Figura 63. Configuración de la plantilla de Ubuntu Server*. Cabe destacar, que el sistema operativo a elegir es ***Other PV (64-bit)*** y no Ubuntu server.

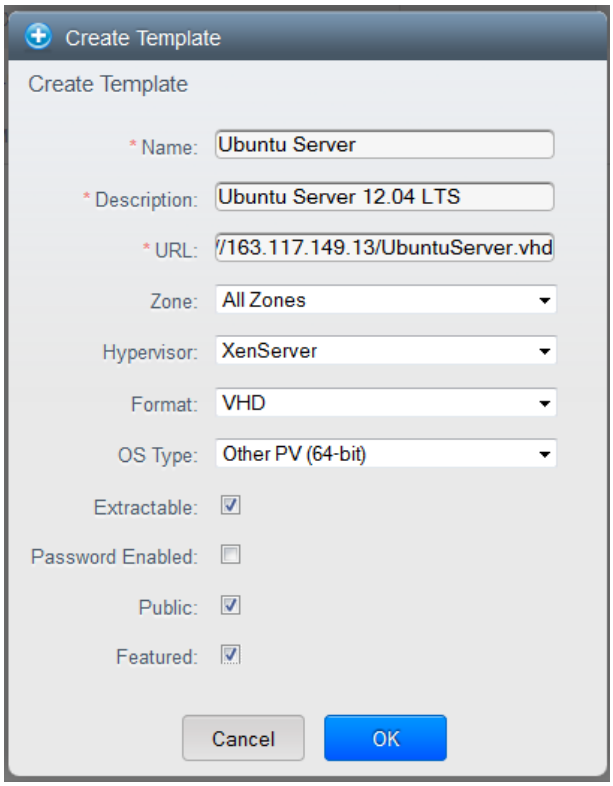


Figura 63. Configuración de la plantilla de Ubuntu Server.

4.2.6. Plantilla Ubuntu Desktop

La instalación de un Ubuntu Desktop es más sencilla. Hay que crear una máquina virtual nueva, seleccionando la plantilla de Windows Server 2008 R2 (64-bit) como plantilla base y, desde el DVD virtual, instalar la imagen de Ubuntu Desktop 11.10 (ubuntu-11.10-desktop-i386.iso), disponible en el repositorio NFS_ISO del sistema de almacenamiento.

La instalación del sistema operativo se realiza siguiendo la configuración típica, añadiendo al usuario cosec como usuario de la máquina virtual con permisos de administrador. Una vez finalizada la instalación del sistema operativo, se efectúa un inicio de sesión para realizar las actualizaciones correspondientes, minimizando así las vulnerabilidades del sistema. Como medidas de seguridad adicionales, se han dispuesto las siguientes instalaciones y configuraciones:

- Instalación de los programas *vim*, *ssh* y *denyhosts* (**sudo apt-get install vim ssh denyhosts**)
- Al igual que se hizo con la plantilla de Ubuntu Server, se configura la clave pública del usuario cosec para que las conexiones a los escritorios se puedan realizar únicamente por intercambio de clave privada y pública. Una vez copiada la clave pública en el escritorio, es preciso crear la carpeta .ssh en el directorio del usuario cosec y copiar el contenido de la clave pública al fichero de claves autorizadas:

cd /home/cosec

mkdir .ssh

cat cosec.pub >> .ssh/authorized_keys

- Adicionalmente, hay que configurar el ***ssh*** para evitar la conexión mediante contraseña, siendo la conexión permitida únicamente mediante certificado. Para ello, se modifica el fichero ***sshd_config*** (**vi /etc/ssh/sshd_config*)*** donde se configuran los siguientes parámetros:

  PubkeyAuthentication yes

  ChallengeResponseAuthentication no

  PasswordAuthentication no

  UsePAM no

- Por último se configura el cortafuegos para que únicamente se puedan realizar conexiones ssh al servidor desde las redes internas del laboratorio web (10.2.0.0/16). El fichero de configuración, así como el procedimiento para su ejecución en el inicio de la máquina, son idénticos a los de la plantilla de Ubuntu Server.

Tras la configurar la plantilla de Ubuntu Desktop, se procede a apagar el equipo para importarlo como plantilla en CloudStack. Para importar el sistema operativo, se realiza la misma operativa que en el caso anterior. Los datos a introducir en el caso de un Ubuntu Desktop son los que se muestran en la *Figura 64. Configuración de la plantilla de Ubuntu Desktop*. Cabe destacar, que el sistema operativo a elegir es **Windows Server 2008 R2 (64-bit)**.

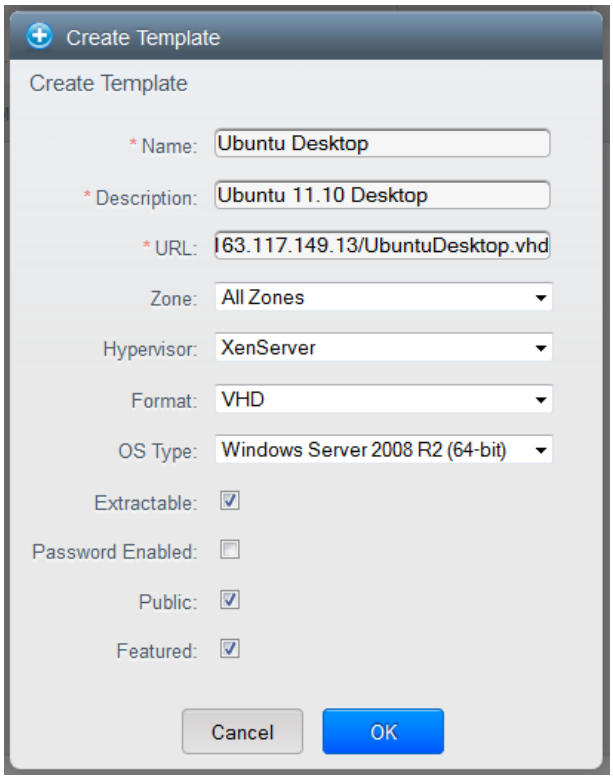


Figura 64. Configuración de la plantilla de Ubuntu Desktop.

4.2.7. Plantilla Windows 7

La siguiente plantilla a configurar será la de Windows 7. Al igual que en casos anteriores, se crea una máquina virtual nueva, seleccionando como plantilla, la de Windows 7 (32-bit) y cargando en el DVD virtual la imagen de Windows 7 (es_windows_7_professional_x86_dvd_x15-65842.iso), disponible en el repositorio NFS_ISO del Abastor.

Al iniciar la máquina virtual, se procede a realizar una instalación típica del sistema operativo, configurando al usuario cosec como usuario administrador de la máquina. Una vez finalizada la instalación del equipo, hay que realizar las actualizaciones de sistema operativo oportunas para así mantener la seguridad del mismo. Además se configuran las actualizaciones del sistema para su descarga e instalación de forma automática en el equipo. Finalizada la instalación y la configuración del sistema operativo, se instalan las herramientas de Xen. Esto permite optimizar el rendimiento del sistema operativo en un entorno virtual.

Antes de proseguir, hay que realizar una operación en el equipo conocida como *sysprep*, que consiste básicamente en eliminar los identificadores del equipo para que cuando se levante una máquina virtual a partir de esta imagen, éstos se regeneren con el objetivo de que no existan equipos duplicados en la red. Dicha configuración evitará a los

usuarios tener que cambiar el nombre del equipo a las nuevas máquinas recién levantadas. El procedimiento llevado a cabo se basa en un artículo de Internet.[42]

Después de finalizar la plantilla de Windows 7, se procede a apagar el equipo para importarlo como plantilla en CloudStack. Para importar el sistema operativo, se siguen los mismos pasos que se siguieron en los casos anteriores Los datos a introducir en el caso de un Windows 7, son los que se muestran en la *Figura 65. Configuración de la plantilla de Windows 7.* Cabe destacar, que el sistema operativo a elegir debería ser **Windows 7 (32-bit)**.

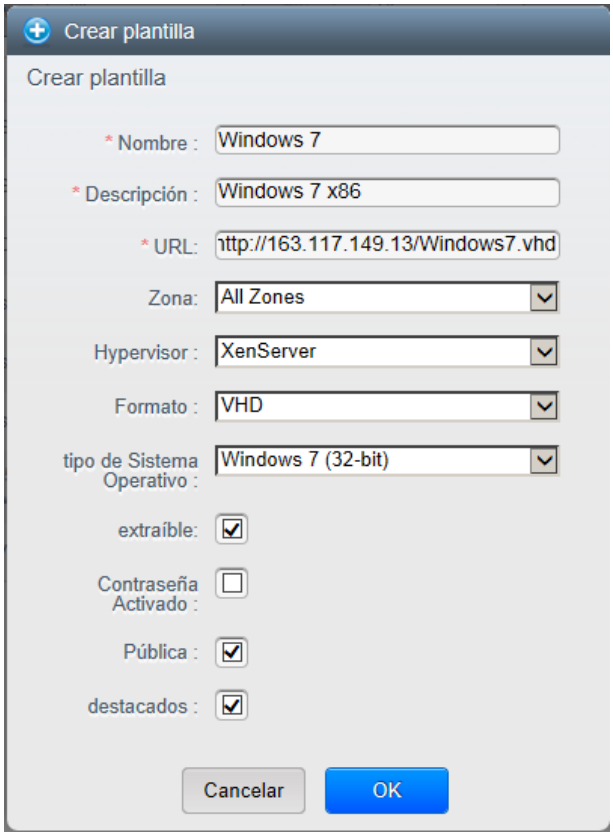


Figura 65. Configuración de la plantilla de Windows 7.

4.2.8. Plantilla Windows XP

Otra de las plantillas que se indican en las especificaciones es la de Windows XP SP3. Al igual que en los casos anteriores, en primer lugar, es necesario crear una máquina virtual con la plantilla por defecto de Windows XP SP3 (32-bit) a través de la consola de Xen. La instalación del sistema operativo se realiza utilizando una imagen iso (es_windows_xp_professional_with_service_pack_3_x86_cd_x14-80488.iso) previamente cargada en el DVD virtual de la máquina.

Posteriormente se actualiza el sistema operativo para que disponga de los últimos parches instalados y se configuran las actualizaciones para que se instalasen de forma

[42] http://theitbros.com/sysprep-a-windows-7-machine-start-to-finish-v2/

automática. Una vez finalizada la instalación y la configuración del sistema operativo es cuando se instalan las herramientas de Xen, lo que permite optimizar el rendimiento del sistema operativo en un entorno virtual. También se crea un usuario, cosec, con permisos de administrador.

Como sucede en el despliegue de equipos Windows, se procede a hacer un sysprep al equipo para eliminar identificadores para que éstos no estén duplicados en la red. El procedimiento a seguir es el indicado en un artículo de Microsoft.[43]

Una vez apagada la máquina virtual, se importa el sistema operativo, siguiendo los pasos ya mencionados Los datos a introducir en el caso de un Windows XP SP3, son los que se muestran en la *Figura 66. Configuración de la plantilla de Windows XP*. Cabe destacar, que el sistema operativo a elegir es **Windows XP SP3 (32-bit)**.

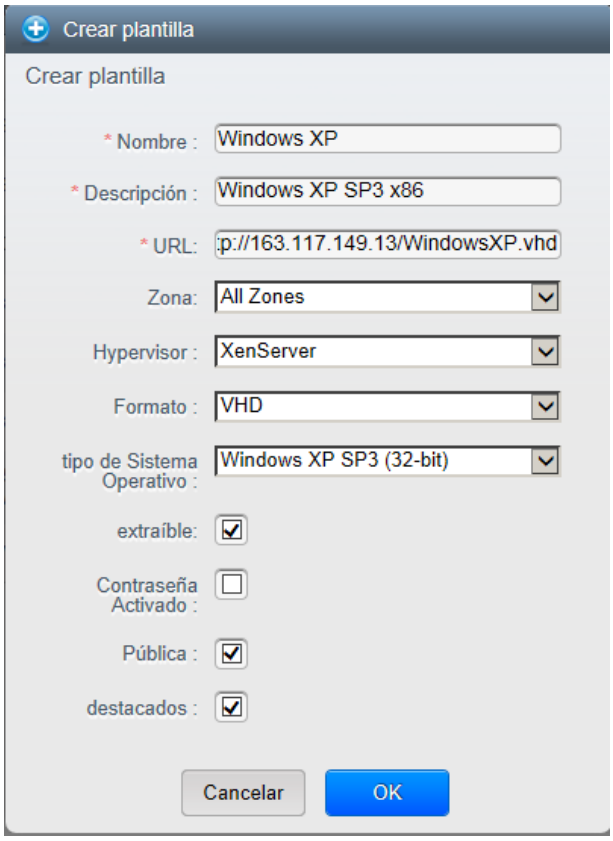


Figura 66. Configuración de la plantilla de Windows XP.

4.2.9. Plantilla Windows Server

La última plantilla configurada para el laboratorio web es la de Windows Server 2008 R2. Su configuración es idéntica al resto de plantillas Windows. También es necesario crear una máquina virtual a partir de una plantilla base, en esta ocasión, Windows Server 2008 R2 (64-bit). En el DVD virtual de la máquina creada se colocar la imagen de instalación del sistema operativo previamente descargada en el servidor de

[43] http://support.microsoft.com/kb/302577/es

almacenamiento (es_windows_server_2008_r2_with_sp1_x64_dvd_617398.iso). Cuando se inicie la máquina virtual, comienza a instalarse el sistema operativo, que se realiza de forma estándar. Una vez finalizada su instalación, es necesario actualizar el sistema operativo para que esté al último nivel de parches. También es necesario instalar las herramientas administrativas de Xen para así optimizar la gestión que hace el hipervisor del equipo.

Como sucede en el despliegue de equipos Windows, es necesario realizar un *sysprep* al equipo para eliminar identificadores y que éstos no estén duplicados en la red. El procedimiento a seguir es el indicado en un artículo de Internet.[44]

Apagada la máquina virtual, se importa el sistema operativo como en los casos anteriores Los datos a introducir en el caso de un Windows server 2008 R2, son los que se muestran en la *Figura 67. Configuración de la plantilla de Windows Server 2008.* Cabe destacar, que el sistema operativo a elegir es **Windows Server 2008 R2 (64-bit)**.

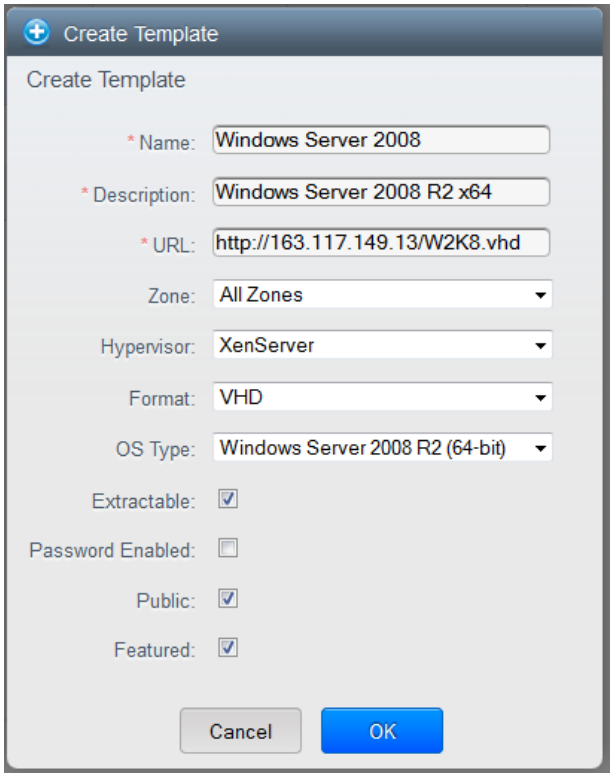


Figura 67. Configuración de la plantilla de Windows Server 2008.

4.2.10. Configuración https

Por defecto, el acceso al interfaz web del laboratorio se realiza a través de *http* al puerto 8080. Para mejorar la seguridad del entorno, se opta por configurar el acceso únicamente a través de *https* y por el puerto 443. El puerto 80 está abierto pero se

[44] http://briandesmond.com/blog/how-to-sysprep-in-windows-server-2008-r2-and-windows-7/

realizará una redirección de http a https. Los pasos a realizar para implementar el cambio, son:

- Efectuar la conexión a la consola del servidor web y, por línea de comandos, abrir el fichero server.xml→ **vi /etc/cloud/management/server.xml**.
- Modificar la sección donde se establece la conexión por el puerto 8080 para que quede se establezca por el puerto 80 y éste se redirija al puerto 443:

```
<Connector executor="tomcatThreadPool"
    port="80" protocol="org.apache.coyote.http11.Http11NioProtocol"

    connectionTimeout="20000" disableUploadTimeout="true"
    acceptCount="150" enableLookups="false" maxThreads="150"
    maxHttpHeaderSize="8192" redirectPort="443" />
 -->
```

- Eliminar los comentarios de la sección que habilita https y modificar el puerto para que quede como se muestra a continuación:

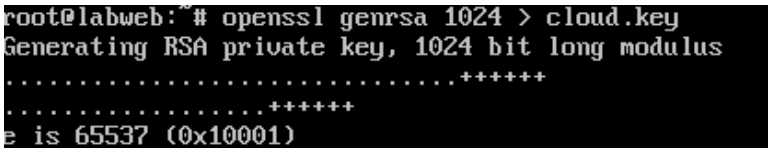
```
<Connector port="443" protocol="HTTP/1.1" SSLEnabled="true"
        maxThreads="150" scheme="https" secure="true"
        clientAuth="false" sslProtocol="TLS"
        keystoreType="PKCS12"
        keystoreFile="conf/cloud-localhost.pk12"
        keystorePass="password"
/>
```

- Modificar el conector para acceder por el Puerto 443:

```
<!-- Define an AJP 1.3 Connector on port 20400 -->
<Connector port="20400" protocol="AJP/1.3" redirectPort="443" />
```

- Crear el certificado oportuno. Los pasos a seguir para crear un certificado válido para el laboratorio son:
    - En primer lugar, generar una clave privada:

      openssl genrsa 1024 > cloud.key

    - En segundo lugar, realizar la solicitud de firma del certificado:

      openssl req -new -key cloud.key > cloud.csr

      Los datos a introducir para la generación de la solicitud son los siguientes:

```
root@labweb:~# openssl req -new -key cloud.key > cloud.csr
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:ES
State or Province Name (full name) [Some-State]:MADRID
Locality Name (eg, city) []:LEGANES
Organization Name (eg, company) [Internet Widgits Pty Ltd]:UC3M
Organizational Unit Name (eg, section) []:COSEC
Common Name (eg, YOUR name) []:www1.seg.inf.uc3m.es
Email Address []:

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:seti:Uc3m
An optional company name []:UC3M
```

- Por último, generar un certificado auto firmado como se muestra a continuación:

  openssl x509 -req -in cloud.csr -signkey cloud.key > cloud.crt

```
root@labweb:~# openssl x509 -req -in cloud.csr -signkey cloud.key > cloud.crt
Signature ok
subject=/C=ES/ST=MADRID/L=LEGANES/O=UC3M/OU=COSEC/CN=www1.seg.inf.uc3m.es
Getting Private key
```

- Crear un almacén de claves en formato *PKCS12* mediante la clave privada y el certificado firmado generados anteriormente:

  openssl pkcs12 -export -in cloud.crt -inkey cloud.key -name cloud -passout pass:password > /usr/share/cloud/management/conf/cloud-localhost.pk12

- Reiniciar el servicio de CloudStack para aplicar los cambios realizados:

  service cloud-management restart

- Uno de los requisitos consiste en redirigir el tráfico del puerto 80 (http) al puerto 443 (https) para tener la posibilidad de acceder a la web por ambos protocolos, pero siendo el que oferta el servicio el puerto seguro (443). Esta configuración se hace en el fichero de **web.xml** (**vi /etc/cloud/management/web.xml**), introduciendo el siguiente código al final del fichero, antes de finalizar la configuración web (</web-app>):

```
<!-- SSL settings. only allow HTTPS access to COSEC -->
<security-constraint>
        <web-resource-collection>
                <web-resource-name>Entire Application</web-resource-name>
                <url-pattern>/*</url-pattern>
        </web-resource-collection>
        <user-data-constraint>
                <transport-guarantee>CONFIDENTIAL</transport-guarantee>
        </user-data-constraint>
</security-constraint>
```

- También se realiza una modificación en el servidor web para que simplemente haya que introducir la url del servidor (http o https) sin que sea necesario introducir la subcarpeta ***/client*** como viene configurado por defecto. Para hacer esto posible, se crea en la carpeta ***/usr/share/cloud/management/conf/Catalina/localhost/*** el fichero ***ROOT.xml*** con el siguiente contenido:

```
<Context
        docBase="/usr/share/cloud/management/webapps"
        path=""
        reloadable="true"
/>
```

4.2.11. Configuraciones globales del sistema.

Como es evidente, es posible, y necesario, la configuración de políticas globales. Unas, únicamente son configurables desde este apartado, mientras que, otras, conviene configurarlas de forma global, para evitar que haya que repetir manualmente la configuración en cada cuenta o proyecto.

Las configuraciones globales se realizan vía web con un usuario administrador del sistema. Una vez se inicia sesión en CloudStack, se navega por el panel izquierdo presionando sobre la opción de configuraciones globales o *Global Settings*.

Figura 68. Botón de configuración de los parámetros globales de CloudStack.

Una vez en la sección, se modifican los parámetros que se consideren oportunos. Seguidamente se describen las configuraciones globales establecidas en este proyecto:

4.2.11.1. Configuración de alertas de seguridad

Uno de los requisitos del entorno es que tenga la capacidad de alertar a los administradores cuando exista algún problema en el laboratorio. Para ello, el sistema está dotado de la utilidad de enviar un correo electrónico a los usuarios que se especifiquen para que estén informados inmediatamente de cualquier posible problema.

Una vez en la sección de configuración global, es necesario modificar los parámetros de envío de alertas. En concreto, las opciones a modificar son:

- ***alert.email.addresses*** → Se especifican las direcciones de correo electrónico a las que se quiera enviar los correos de alerta. En caso de introducir más de una dirección de correo electrónico, las direcciones se separarán por coma. Las direcciones de correo electrónico configuradas son:

  guillermo.suarez.tangil@uc3m.es

- ***alert.email.sender*** → Mediante ella se configura la dirección origen del correo electrónico. No es necesario que sea una dirección real. Es un dato informativo. La dirección configurada es: ***CosecCloudStack@uc3m.es***
- ***alert.smtp.host*** → En este parámetro, se configura la dirección del servidor smtp[45] para el envío del correo electrónico de alerta. Se utiliza el servidor *smtp* de la Universidad: ***smtp.uc3m.es***
- ***alert.smtp.port*** → En este parámetro se configura el puerto configurado en el servidor smtp para el envío de correos electrónicos. El puerto utilizado es el ***25*** como se indica en la url de la Universidad dedicada al servicio de correo electrónico[46].
- ***alert.wait*** → Este último parámetro establece el tiempo de espera entre el momento que se genera la alerta y el momento en el que se envía el correo electrónico a los administradores. El tiempo se establece en **5** segundos.

alert.email.addresses	Comma separated list of email addresses used for sending alerts.	guillermo.suarez.tangil@uc3m.es	
alert.email.sender	Sender of alert email (will be in the From header of the email).	CosecCloudStack@uc3m.es	
alert.smtp.host	SMTP hostname used for sending out email alerts.	smtp.uc3m.es	
alert.smtp.password	Password for SMTP authentication (applies only if alert.smtp.useAuth is true).		
alert.smtp.port	Port the SMTP server is listening on.	25	

Figura 69. Configuración del envío de alertas.

4.2.11.2. Configuración de la descarga de plantillas e imágenes

Uno de los requisitos establecidos en la generación del entorno web es la posibilidad de descargarse cualquier imagen en formato ISO de Internet. Para que esto sea posible, es necesario modificar un parámetro de la configuración de CloudStack que permite la descarga de estos ficheros desde cualquier IP. El parámetro, ***secstorage.allowed.internal.sites***, se define como 0.0.0.0/0, para que se puedan realizar descargas de cualquier dirección web de Internet. Para configurar el parámetro, hay que ir a Configuraciones Globales, o *Global Settings*, localizar la variable *global secstorage.allowed.internal.sites*, y modificarla para que su valor sea 0.0.0.0/0 como se muestra en la siguiente imagen:

[45] Protocolo de envío simple de correos electrónicos o *Simple Mail Transfer Protocol.* Se trata de un protocolo de red utilizado para el intercambio de mensajes de correo electrónico. http://es.wikipedia.org/wiki/Simple_Mail_Transfer_Protocol

[46] http://portal.uc3m.es/portal/page/portal/informatica/NosDedicamos/ServiciosCorporativos/CorreoElectronico

secstorage.allowed.internal.sites	Comma separated list of cidrs internal to the datacenter that can host template download servers	0.0.0.0/0

Figura 70. Configuración de descarga de imágenes ISO.

4.2.11.3. Configuración de la liberación de recursos

Por defecto, las máquinas virtuales, cuentas, proyectos…que se eliminan, permanecen en la base de datos y por tanto en el sistema, durante 48 horas. En nuestro caso, se realiza una modificación a nivel global para disminuir el tiempo a 12 horas, ya que se considera que 12 horas es un tiempo más que suficiente para rectificar en la eliminación de un recurso. Los parámetros modificados son los siguientes:

account.cleanup.interval	The interval (in seconds) between cleanup for removed accounts	43200	
expunge.delay	Determines how long (in seconds) to wait before actually expunging destroyed vm. The default value = the default value of expunge.interval	43200	
expunge.interval	The interval (in seconds) to wait before running the expunge thread.	43200	
storage.cleanup.interval	The interval (in seconds) to wait before running the storage cleanup thread.	43200	
vm.op.cleanup.interval	Interval to run the thread that cleans up the vm operations (in seconds)	43200	

Figura 71. Configuración de liberación de recursos.

4.2.11.4. Configuración de cuentas y proyectos

Por defecto, por cada cuenta y proyecto puede crearse 20 redes con salida a Internet. En el entorno diseñado, los recursos son limitados y, salvo el administrador, ningún usuario o proyecto debería tener la capacidad de utilizar IPs públicas. Para obtener el resultado esperado, hay que cambiar algunos valores, que se muestran en la siguiente figura:

max.account.networks	The default maximum number of networks that can be created for an account	0	
max.account.public.ips	The default maximum number of public IPs that can be consumed by an account	0	
max.project.networks	The default maximum number of networks that can be created for a project	0	
max.project.public.ips	The default maximum number of public IPs that can be consumed by a project	0	

Figura 72. Configuración de cuentas y proyectos.

4.2.11.5. Configuración máxima de disco de datos.

Para evitar la creación de discos de datos de tamaño considerable se limita el tamaño máximo de los discos de datos a 50 GB y a un número máximo de éstos, de 10 por cuenta y de 10 por proyecto. Para ello, a continuación se pueden apreciar las variables modificadas:

max.project.volumes	The default maximum number of volumes that can be created for a project	10	
max.account.volumes	The default maximum number of volumes that can be created for an account	10	
storage.max.volume.size	The maximum size for a volume (in GB).	50	

Figura 73. Configuración de los discos de datos.

4.2.12. Oferta de Servicios.

Se establecen el conjunto de servicios ofertados por el laboratorio web para gestionar el entorno de forma eficiente y permitiendo los suficientes recursos a los usuarios para el desarrollo de los diferentes proyectos.

4.2.12.1. Oferta de Cómputo

Se definen cuatro servicios de cómputo. Cada servicio de cómputo se crea por el administrador del sistema siguiendo los siguientes pasos y estableciendo los parámetros correspondientes para cada uno de los servicios de cómputo. Para ello, en primer lugar, es necesario iniciar sesión en CloudStack. Una vez iniciada sesión se presiona sobre el icono de Oferta de Servicios o *Service Offerings* que aparece en el panel izquierdo.

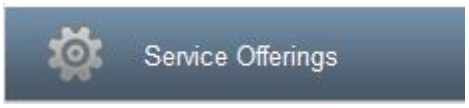


Figura 74. Botón de oferta de servicios de CloudStack.

Abierta la pantalla de oferta de servicios, hay que seleccionar el servicio de cómputo o *Compute offerings*:

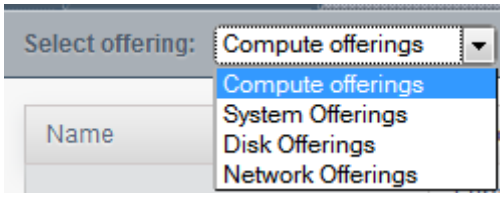


Figura 75. Selección del servicio de cómputo.

Se añade uno nuevo presionando el botón *Add compute offering*, tras lo que aparece la pantalla correspondiente donde hay que introducir los datos oportunos para cada tipo de instancia:

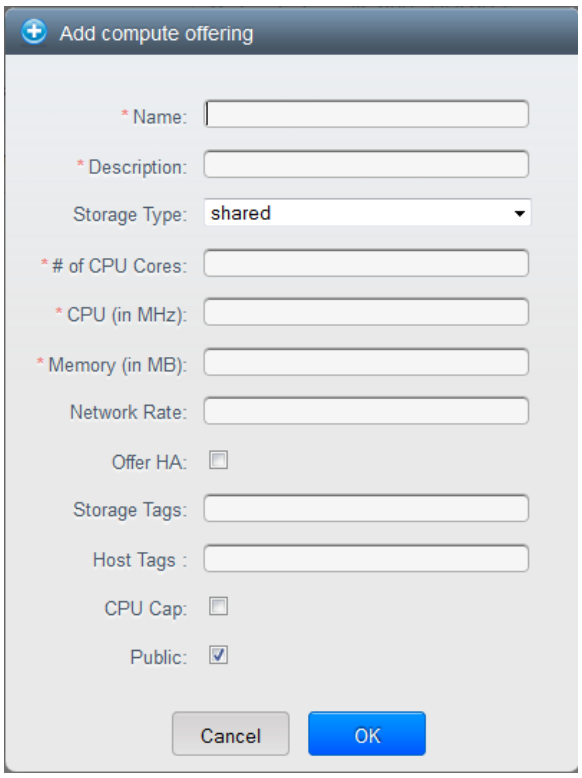


Figura 76. Configuración del servicio de cómputo.

Una vez se presione sobre el botón de *OK*, el servicio está disponible para los usuarios. A continuación se muestran los servicios de cómputo configurados:

Name	Description	Order
Large Instance	Large Instance (2 vCPUs (1000 MHz) + 2048 MB RAM)	
Medium Instance	Medium Instance (2 vCPUs (750 MHz) + 1024 MB RAM)	
Small Instance	Small Instance (1 vCPU (750 MHz) + 768 MB RAM)	
Tiny Instance	Tiny Instance (1 vCPU (500 MHz) + 512 MB RAM)	

Figura 77. Servicios de cómputo configurados.

4.2.12.2. Oferta de Disco

Se definen cuatro tipos de disco. Para su configuración, una vez iniciada sesión en CloudStack, se presiona sobre el icono de Oferta de Servicios o *Service Offerings* que aparece en el panel izquierdo.

Figura 78. Botón de oferta de servicios de CloudStack-

Una vez en la pantalla de oferta de servicios, se selecciona el servicio de disco o *Disk offerings*:

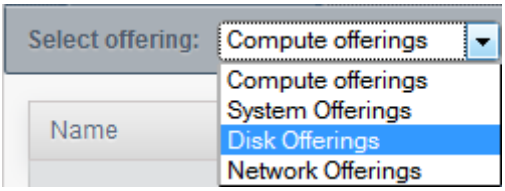


Figura 79. Selección del servicio de disco.

Y se añade uno nuevo dando al botón *Add Disk Offering*. Aparece la pantalla correspondiente donde hay que introducir los datos oportunos para cada tipo de disco:

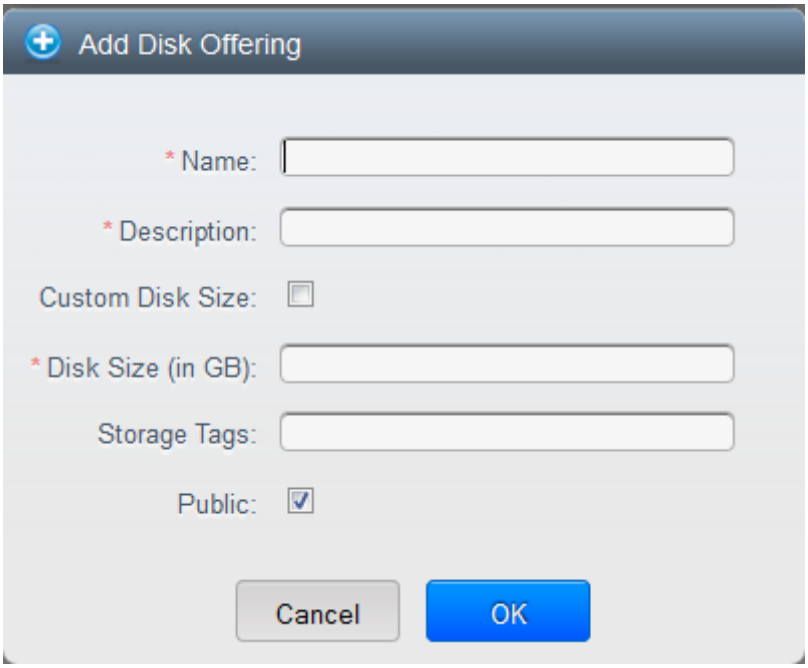


Figura 80. Configuración del servicio de disco.

Una vez se presione sobre el botón de *OK*, el servicio está disponible para los usuarios. A continuación se muestran los discos configurados:

Name	Description	Custom Disk Size	Disk Size (in GB)	Order
Small	Small Disk, 5 GB	No	5	
Medium	Medium Disk, 20 GB	No	20	
Custom Disk	Custom Disk	Yes	N/A	

Figura 81. Servicios de disco configurados.

4.2.13. Problemas y Consejos

Esta sección describe alguno de los problemas encontrados a lo largo de la realización del proyecto y sus posibles soluciones o consejos de acción.

4.2.13.1. Máquinas se quedan en estado Stopping

Puede darse el caso, de manera esporádica, que una máquina se quede en estado *stopping* o parando. Este problema se puede producir cuando se reinicia una máquina virtual y, de forma simultánea, se reinicia el servidor de gestión o WebFrontend. La máquina, aun estando apagada se queda constantemente en estado *stopping*, por lo que queda no operativa. Para solucionar este problema, basta con modificar su estado en la base de datos:

mysql -uroot -p (Tras introducir este comando, es necesario introducir la contraseña del usuario root de la base de datos)

use cloud; (Selección de la bbdd de CloudStack)

select id,name,state from vm_instance; (Este comando muestra todas las máquinas virtuales y su estado)

UPDATE vm_instance SET state='Stopped' WHERE state='Stopping'; (Este comando actualiza la base de datos, modificando el estado de las máquinas virtuales de stopping a stopped)

4.2.13.2. Servidores Ubuntu no arrancan

A lo largo de la realización del proyecto, se ha producido el caso en el que los servidores Ubuntu no arrancan al realizarse una actualización de los paquetes de sistema. Esto se debe a que la actualización del grub[47] (sistema de arranque de Linux) en los sistemas hace que éstos sean incapaces de iniciar en Xen. Para evitar esta incidencia, se recomiendan dos soluciones. Una es una solución al problema y otra propone evitar el mismo.

- Deshabilitar la actualización del grub en los sistemas Ubuntu Server incluyendo la actualización del grub en una lista *negra* para que no se actualice nunca.
- Una vez actualizado el grub, habría que seguir los pasos que se indican en el siguiente documento para reparar el grub actualizado. http://www.virtualzone.de/2012/06/ubuntu-vm-not-starting-on-xenserver.html

[47] https://en.wikipedia.org/wiki/GNU_GRUB

Capítulo 5

5. EVALUACIÓN

En este capítulo se describe la realización de una serie de experimentos para evaluar el funcionamiento del sistema y detectar los posibles puntos de fallo o mejoras en el mismo. Para ello, en primer lugar se expone el desarrollo completo de un experimento para, posteriormente, presentar los resultados derivados de la ejecución de una batería de experimentos.

5.1. Funcionamiento de un experimento y reglas aplicadas

Para entender el funcionamiento de un experimento genérico, y describir las reglas aplicadas, se utiliza un ejecutable que el sistema detecta como malware. En este caso, el ejecutable utilizado es el fichero *Worm.Win32.AutoRun.dvq.exe*. El experimento base consiste en lanzar el laboratorio con 2 máquinas virtuales de cada tipo y una duración del experimento de treinta minutos. Durante la ejecución, una de las máquinas virtuales de cada modelo ejecuta la pieza de malware mientras que la otra no lo hace: simplemente realiza sus operativas normales.

En el caso descrito, el comando necesario para su ejecución es:

```
./lab_auto.sh –t 30m –n 2 –e Worm.Win32.AutoRun.dvq.exe
```

Las reglas aplicadas al entorno se muestran a continuación y se configuran en el fichero *ossec.conf*:

Se monitorizan los siguientes directorios:

- C:\WINDOWS/System32
- C:\WINDOWS/SysWOW64
- C:\Archivos de programa
- C:\Archivos de programa (x86)
- %WINDIR%\ssms.exe
- %WINDIR%\version.txt
- %WINDIR%\winhelp32.exe

- %WINDIR%\winlogon.exe
- %WINDIR%/win.ini
- %WINDIR%/system.ini
- C:\autoexec.bat
- C:\config.sys
- C:\boot.ini
- %WINDIR%/regedit.exe
- %WINDIR%/explorer.exe
- %WINDIR%/notepad.exe
- %WINDIR%/taskman.exe
- C:\Documents and Settings/Administrador/MENINI~1/Programas/Inicio
- C:\Documents and Settings/All Users/MENINI~1/Programas/Inicio
- C:\Users/uc3m/AppData/Roaming/Microsoft/Windows/STARTM~1/Programas/Startup

Y las siguientes claves de registro:

- HKEY_LOCAL_MACHINE\Software\Classes\batfile
- HKEY_LOCAL_MACHINE\Software\Classes\cmdfile
- HKEY_LOCAL_MACHINE\Software\Classes\comfile
- HKEY_LOCAL_MACHINE\Software\Classes\exefile
- HKEY_LOCAL_MACHINE\Software\Classes\piffile
- HKEY_LOCAL_MACHINE\Software\Classes\AllFilesystemObjects
- HKEY_LOCAL_MACHINE\Software\Classes\Directory
- HKEY_LOCAL_MACHINE\Software\Classes\Folder
- HKEY_LOCAL_MACHINE\Software\Classes\Protocols
- HKEY_LOCAL_MACHINE\Software\Policies
- HKEY_LOCAL_MACHINE\Security
- HKEY_LOCAL_MACHINE\Software\Microsoft\Internet Explorer
- HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services
- HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Session Manager\KnownDLLs
- HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\SecurePipeServers\winreg
- HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run
- HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\RunOnce
- HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\RunOnceEx
- HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\URL
- HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\PoliciesHKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\URL
- HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Policies
- HKEY_LOCAL_MACHINE\Software\Microsoft\Windows NT\CurrentVersion\Windows
- HKEY_LOCAL_MACHINE\Software\Microsoft\Windows NT\CurrentVersion\Winlogon
- HKEY_LOCAL_MACHINE\Software\Microsoft\Windows NT\CurrentVersion\Image File Execution Options
- HKEY_LOCAL_MACHINE\Software\Microsoft\Active Setup\Installed Components

Los resultados obtenidos de la ejecución del experimento se resumen a continuación:

Hora inicio experimento: martes 30 de abril 2013 a las 10:29:36

Hora fin creación entorno: martes 30 de abril 2013 a las 10:49:38 (20’ 02’’)

Los equipos iniciados y aplicaciones instaladas se muestran en la siguiente tabla:

Aplicación	WXPSP2_1	WXPSP2_2	WXPSP3_1	WXPSP3_2	W2K3_1	W2K3_2	W7SP1_1	W7SP1_2	W2K8_1	W2K8_2
Adobe Reader 9.1	✔	✔	✔			✔	✔	✔		
Adobe Reader 10.1.0				✔	✔					
Adobe Flash Player 9.0.124	✔							✔		
Adobe Flash Player 10.0.32.18		✔			✔					✔
Adobe Flash Player 10.3.183.7			✔	✔		✔				
Mozilla Firefox 6.0.2	✔	✔	✔		✔			✔		
Mozilla Firefox 13.0.1				✔		✔				
K-Lite Codec pack 5.1.0			✔		✔					
K-Lite Codec pack 9.2.0	✔	✔		✔		✔		✔		
.NET Framework 1.1 SP1	✔	✔	✔	✔	✔	✔		✔		
.NET Framework 2.0 SP2	✔	✔			✔			✔		
.NET Framework 3.5 SP1	✔	✔	✔		✔	✔		✔		
Java 1.4.2.07	✔	✔	✔	✔	✔	✔		✔		
Java 1.5.0.11	✔	✔		✔	✔			✔		
Java 1.6.0.70	✔	✔		✔	✔	✔				
MSXML 4 SP2	✔			✔		✔		✔		
MSXML 6	✔	✔		✔	✔					
QuickTime 6.5	✔	✔						✔		
QuickTime 7.64.17.73				✔	✔					
Macromedia Shockwave 9					✔	✔				
Macromedia Shockwave 11	✔	✔	✔	✔				✔		✔
Google Chrome 21.0.1180	✔	✔			✔	✔		✔		

Tabla 43. Aplicaciones instaladas en el experimento de referencia.

Hora inicio ejecución pieza de malware: martes 30 de abril 2013 a las 10:50:18

Tiempo ejecución experimento aislado: 30’

Hora inicio eliminación laboratorio: martes 30 de abril 2013 a las 11:20:18

Hora fin eliminación laboratorio: martes 30 de abril 2013 a las 11:22:02

Tiempo eliminación laboratorio: 1’ 44’’

Tiempo total del experimento: 52’ 26’’

Los resultados obtenidos de la ejecución del experimento son los que se muestran a continuación:

Worm.Win32.AutoRun.dvq.exe	Cantidad	% sobre total
# Total Eventos	349	100%
# Agentes iniciados	14	4%
# Eventos auditoría Windows	40	11.5%
# Errores Windows	49	14%
# Malware	1	0.3%
# Cambios de integridad	146	41.8%
# Ficheros añadidos	1	0.3%
# Eventos de cambio de hora	1	0.3%
# Eventos inicio de sesión	44	12.6%
# Eventos cierre de sesión	0	0%

Tabla 44. Tabla con los resultados del experimento de referencia.

Toda la información está recogida en el fichero *config_experimento.txt* generado en la ejecución de cada experimento. Este fichero aporta la información correspondiente a las horas de inicio y fin de las principales operaciones que acontecen en el experimento así como un resumen final de los diferentes eventos producidos.

5.2. Batería de experimentos

Una vez visto el funcionamiento genérico de un experimento, la ejecución de cualquier otro experimento tiene en consideración los mismos parámetros anteriormente comentados, variando únicamente el fichero que se va a ejecutar. Obviamente, también varían las aplicaciones instaladas (al instalarse de forma aleatoria) en los diferentes sistemas operativos y los resultados obtenidos de la ejecución de cada uno de ellos. Los ejecutables se han seleccionado de forma específica para que no exista la posibilidad de infectar al hipervisor.

A continuación se muestra, en la *Tabla 45. Resumen de la ejecución de experimentos*., los resultados obtenidos de la ejecución de varios experimentos en el entorno aislado. En ella se destacan los resultados que se consideran más relevantes para su análisis.

Además, se incluyen gráficas comparativas entre los diferentes experimentos.

Ejecutable	Tiempo total	#Total eventos	#Eventos auditoría Windows	#Errores Windows	Malware	#Cambios checksum	#Ficheros nuevos	Cambios de hora	#Logon	#Logoff
AutoRun.dvq	52’ 26’’	349 (100%)	40 (11.5%)	49 (14%)	1 (0.3%)	146 (41.8%)	1 (0.3%)	1 (0.3%)	44 (12.6%)	0 (0%)
AutoRun.obt	49’ 45’’	202 (100%)	47 (23.3%)	11 (5.5%)	0 (0%)	33 (16.3%)	0 (0%)	2 (1%)	42 (20.8%)	0 (0%)
AutoRun.obz	46’ 25’’	307 (100%)	45 (14.7%)	10 (3.3%)	0 (0%)	122 (39.8%)	1 (0.3%)	2 (0.7%)	58 (18.9%)	15 (4.9%)
AutoRun.dvl	47’ 08’’	179 (100%)	46 (25.7%)	13 (7.3%)	0 (0%)	16 (8.9%)	0 (0%)	2 (1.1%)	41 (22.9%)	0 (0%)
AutoRun.dvf	45’ 47’’	194 (100%)	42 (21.7%)	9 (4.6%)	0 (0%)	24 (12.4%)	1 (0.5%)	1 (0.5%)	54 (27.8%)	5 (2.6%)
AutoRun.eba	47’ 48’’	198 (100%)	53 (26.8%)	16 (8.1%)	0 (0%)	27 (13.6%)	1 (0.5%)	2 (1%)	44 (22.2%)	0 (0%)
AutoRun.dmj	48’ 29’’	1639 (100%)	46 (2.8%)	12 (0.7%)	0 (0%)	19 (1.2%)	1 (0.1%)	1463 (89.3%)	43 (2.6%)	0 (0%)
AutoRun.obq	47’ 04’’	305 (100%)	49 (16.1%)	24 (7.9%)	0 (0%)	137 (45%)	0 (0%)	0 (0%)	41 (13.4%)	0 (0%)
AutoRun.dvg	50’ 09’’	219 (100%)	48 (21.9%)	12 (5.5%)	0 (0%)	22 (10.1%)	3 (1.4%)	1 (0.5%)	61 (27.9%)	16 (7.3%)
AutoRun.ebx	51’ 48’’	220 (100%)	50 (22.7%)	16 (7.3%)	0 (0%)	37 (16.8%)	6 (2.7%)	1 (0.5%)	46 (20.9%)	0 (0%)
AutoRun.dmd	44’ 14’’	177082 (100%)	157 (0.1%)	8 (0%)	0 (0%)	14 (0%)	1 (0%)	1 (0%)	28515 (16.1%)	8368 (4.7%)
AutoRun.ebj	51’ 47’’	235 (100%)	56 (23.83%)	15 (6.4%)	0 (0%)	20 (8.5%)	1 (0.4%)	2 (0.8%)	59 (25.1%)	15 (6.4%)
AutoRun.obd	45’ 28’’	224 (100%)	52 (23.2%)	10 (4.5%)	0 (0%)	32 (14.3%)	1 (0.5%)	1 (0.5%)	58 (25.9%)	16 (7.1%)

Ejecutable	Tiempo total	#Total eventos	#Eventos auditoría Windows	#Errores Windows	Malware	#Cambios checksum	#Ficheros nuevos	Cambios de hora	#Logon	#Logoff
AutoRun.cje	49’ 49’’	1599 (100%)	45 (2.8%)	14 (0.9%)	0 (0%)	52 (3.3%)	0 (0%)	1418 (88.7%)	40 (2.5%)	0 (0%)
AutoRun.cjr	50’ 48’’	1214 (100%)	52 (4.3%)	16 (1.3%)	0 (0%)	1025 (84.4%)	7 (0.6%)	1 (0.1%)	45 (3.7%)	0 (0%)
AutoRun.ebi	49’ 30’’	1794 (100%)	43 (2.4%)	62 (3.5%)	0 (0%)	1526 (85.1%)	52 (2.9%)	0 (0%)	49 (2.7%)	0 (0%)
AutoRun.vpy	48’ 53’’	252 (100%)	68 (27%)	16 (6.4%)	0 (0%)	41 (16.3%)	0 (0%)	1 (0.4%)	50 (19.8%)	0 (0%)
AutoRun.vq	50’ 53’’	186 (100%)	49 (26.3%)	10 (5.4%)	0 (0%)	20 (10.8%)	0 (0%)	0 (0%)	45 (24.2%)	0 (0%)
AutoRun.vr	48’ 25’’	219 (100%)	42 (19.2%)	7 (3.2%)	0 (0%)	20 (9.1%)	2 (0.9%)	4 (1.8%)	66 (30.1%)	16 (7.3%)
AutoRun.nxi	48’ 49’’	1665 (100%)	50 (3%)	16 (1%)	0 (0%)	13 (0.8%)	1 (0.1%)	1474 (88.5%)	39 (2.3%)	0 (0%)
AutoRun.zkc	47’ 30’’	224 (100%)	44 (19.6%)	8 (3.6%)	0 (0%)	28 (12.5%)	8 (3.6%)	4 (1.8%)	63 (28.1%)	15 (6.7%)
AutoRun.zky	48’ 31’’	315 (100%)	114 (36.2%)	83 (26.4%)	0 (0%)	20 (6.4%)	1 (0.3%)	4 (1.3%)	42 (13.3%)	0 (0%)
AutoRun.oi	48’ 37	227 (100%)	57 (25.1%)	7 (3.1%)	0 (0%)	10 (4.4%)	0 (0%)	4 (1.8%)	62 (27.3%)	7 (3.1%)
AutoRun.ega	49’ 23’’	8403 (100%)	2963 (35.3%)	10 (0.1%)	0 (0%)	70 (0.8%)	0 (0%)	3 (0%)	2617 (31.1%)	0 (0%)
AutoRun.egf	50’ 17’’	225 (100%)	51 (22.7%)	9 (4%)	0 (0%)	54 (24%)	0 (0%)	4 (1.8%)	45 (20%)	0 (0%)
AutoRun.egg	49’ 47’’	67229 (100%)	14200 (21.1%)	10 (0.01%)	0 (0%)	28 (0.04%)	697 (1%)	4 (0.01%)	17337 (25.8%)	1965 (2.9%)

Ejecutable	Tiempo total	#Total eventos	#Eventos auditoría Windows	#Errores Windows	Malware	#Cambios checksum	#Ficheros nuevos	Cambios de hora	#Logon	#Logoff
AutoRun.ego	48’ 04’’	208 (100%)	42 (20.2%)	8 (3.9%)	0 (0%)	24 (11.5%)	0 (0%)	3 (1.4%)	53 (25.5%)	15 (7.2%)
AutoRun.egh	49’ 27’’	217 (100%)	48 (22.1%)	8 (3.7%)	0 (0%)	20 (9.2%)	0 (0%)	6 (2.8%)	65 (30%)	15 (6.9%)

Tabla 45. Resumen de la ejecución de experimentos.

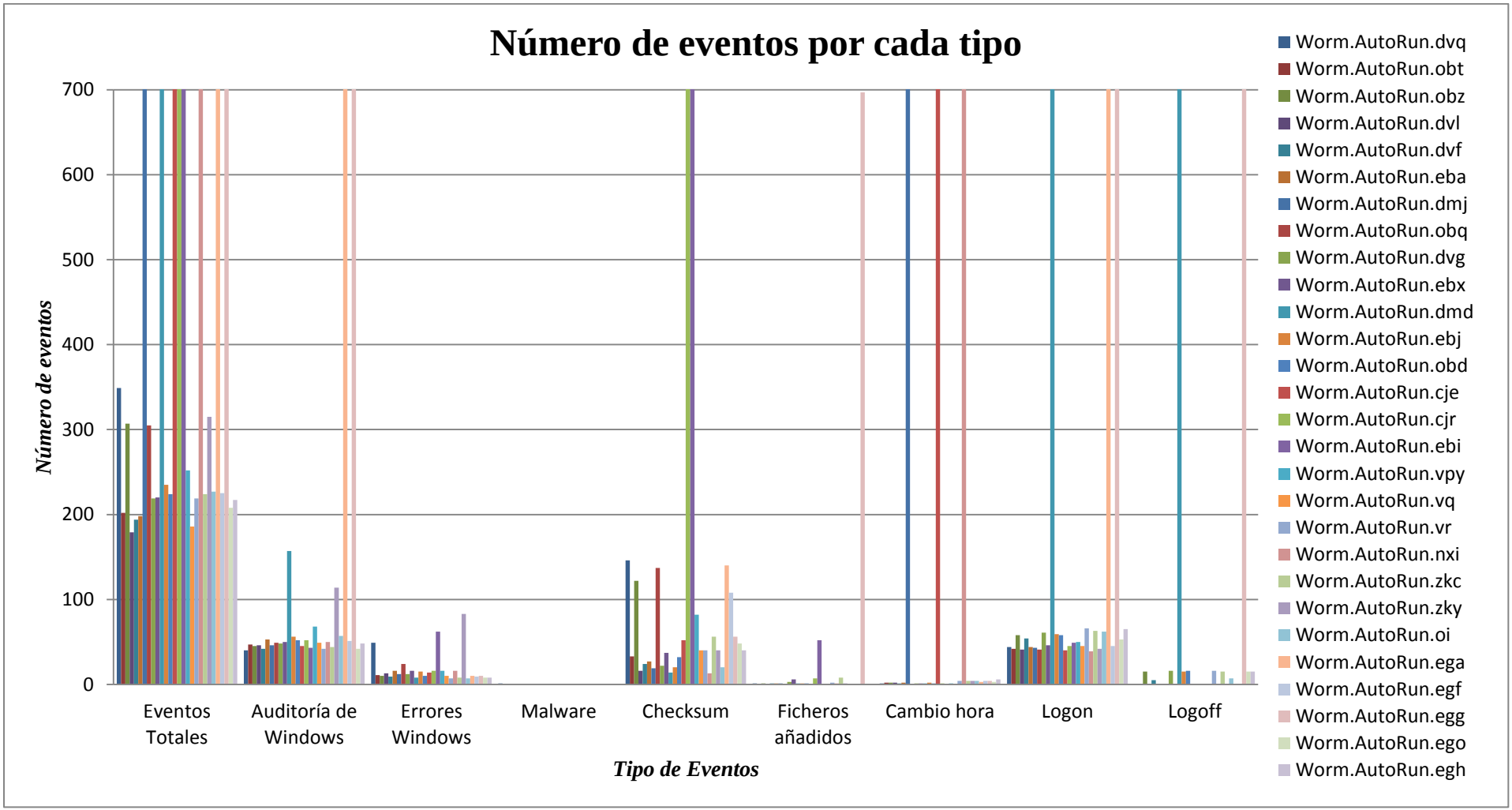


Figura 82. Número de eventos por cada tipo.

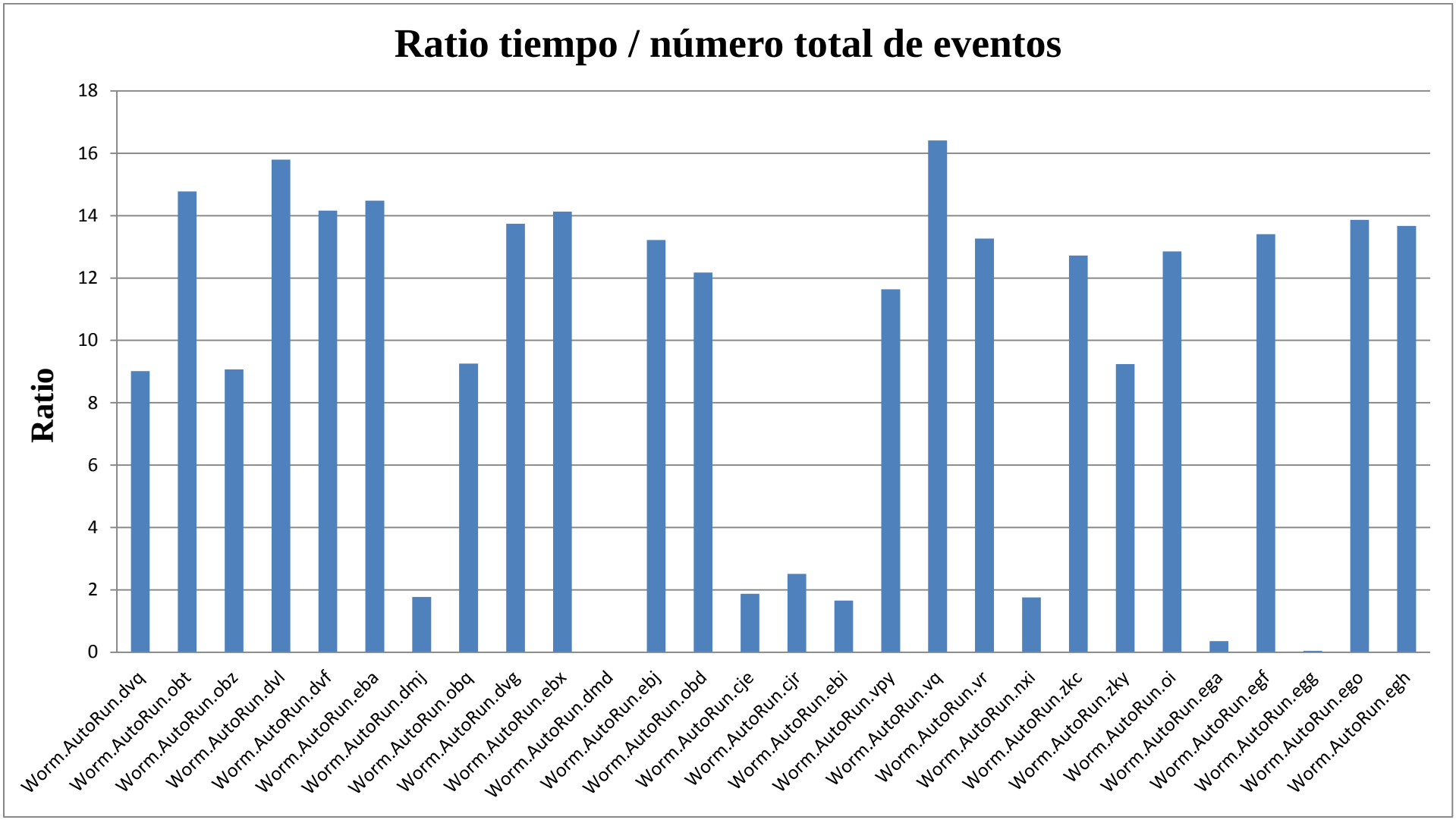


Figura 83. Ratio tiempo / número de eventos

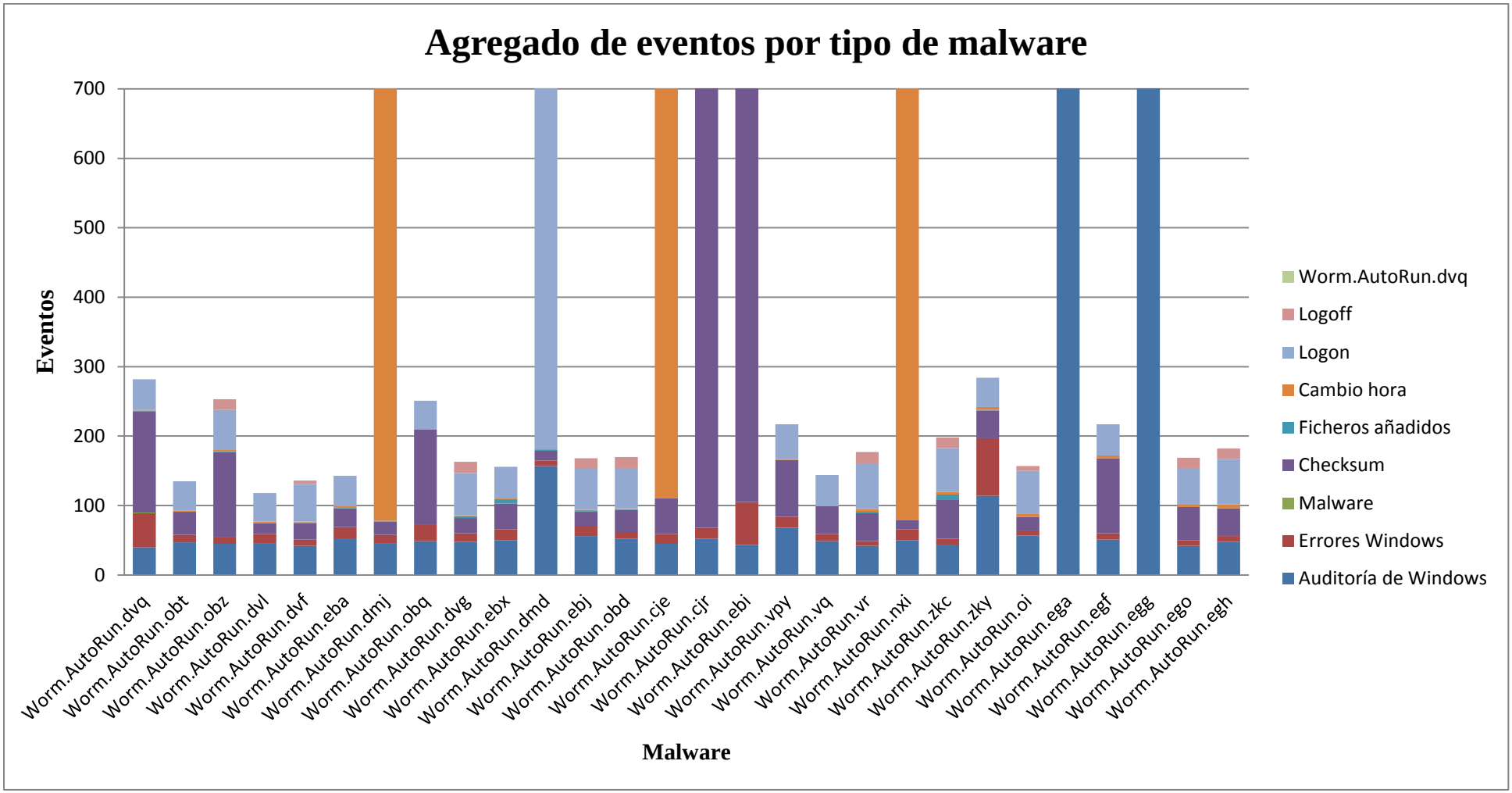


Figura 84. Agregado de eventos por tipo de malware.

En la documentación proporcionada, se adjunta la configuración completa de los experimentos. La estructura de cada experimento consta de las siguientes las siguientes carpetas:

- Carpeta con el número de experimento. El nombre está compuesto por catorce dígitos. Es la carpeta donde se encuentran todos los datos del experimento.
    - Carpeta *machines*, donde se encuentran los ficheros de configuración de cada máquina virtual del experimento.
    - Carpeta *events*. En ésta, se encuentra el fichero con los eventos capturados por el servidor OSSIM.
    - Fichero ***config_experimento.txt*** donde se resume la configuración del experimento. Incluye las horas de ejecución, la configuración de las máquinas y el resumen de los eventos entre otros detalles.

5.2.1.1. Discusión de los resultados

El análisis de la *Tabla 45. Resumen de la ejecución de experimentos*. demuestra con claridad que los experimentos arrojan información sobre los efectos de los ficheros ejecutados en cada experimento. Sin embargo, no se detectan indicios de que el *malware* explote alguna vulnerabilidad propia del software instalado. Es necesario realizar pruebas metódicas para detectar estas relaciones. Se expone en el apartado *7.2*, el estudio de las posibles relaciones entre un malware y las vulnerabilidades de los sistemas.

En algunos casos, como en la ejecución del fichero *Worm.AutoRun.ega*, se produce un elevado número de eventos de inicio de sesión. Esto podría suponer que está intentando conectarse al resto de equipos de la red. En otros experimentos se produce un elevado número de modificaciones en ficheros o claves de registro, como sucede por ejemplo en el experimento donde se ejecuta el fichero *Worm.AutoRun.ebi*…También hay experimentos en los que se aprecia un número elevado de cambios de hora en los sistemas, como en el caso de ejecución del fichero *Worm.AutoRun.dmj*, donde el 89% de los eventos son de este tipo. El análisis de los eventos monitorizados podría servir para caracterizar los diferentes tipos de *malware*.

En el experimento donde se ejecuta el fichero *Worm.AutoRun.dmd*, destaca una cifra elevada en el número total de eventos capturados en comparación con los eventos monitorizados: Se detecta únicamente un 15% de eventos. Un análisis en profundidad del fichero de eventos de dicho experimento destaca la aparición de un evento que no se había tenido en cuenta a la hora de monitorizar el entorno: *Logon Failure*. Este evento supone un 80.4% sobre el número total de eventos, lo que sugiere que el fichero ejecutado intenta realizar un ataque por fuerza bruta a toda la red.

Otro caso a destacar son los resultados de los experimentos en los que se ejecuta ficheros *Worm.Win32.AutoRun.ega* y *Worm.Win32.AutoRun.egg*. En la ejecución de estos experimentos se detecta un evento "*Logon Failure - Unknown user or bad password*" que tampoco se había producido en los experimentos de prueba.

También cabe destacar que durante las pruebas iniciales de ejecución de malware en los entornos aislados, se detectaron una serie de eventos a tener en cuenta. En algunos experimentos, a la hora de ejecutar el fichero correspondiente, aparece un mensaje en el que se indica que la aplicación no se puede ejecutar en entornos Windows. En otros casos, la intrusión consiste en abrir una ventana de Internet Explorar para realizar la conexión a una dirección de Internet de donde presumiblemente intenta efectuar la descarga de un fichero, posiblemente el auténtico *malware*. También existen experimentos en los que se abre un programa con el que hay que interactuar y, en función de esa interacción, los efectos en los equipos son aleatorios.

Capítulo 6

6. GESTIÓN DEL PROYECTO

En este capítulo se ofrece una visión general de la organización y administración de todos los recursos que han formado parte del proyecto, se analiza la metodología utilizada, la planificación inicial y final, llevada a cabo, así como el presupuesto inicial y final del proyecto. También se detallan las herramientas, *software* y hardware, utilizadas para la realización del proyecto.

6.1. Metodología de desarrollo

La metodología utilizada a la hora de realizar cualquier tipo de proyecto resulta de gran importancia para que éste llegue o no, a buen puerto. En el caso particular de la realización de este proyecto, la opción de la metodología a utilizar era vital, no sólo por la importancia del mismo sino porque me veía abocado a realizarlo con recursos escasos ya que, al encontrarme trabajando por cuenta ajena, no sólo me encontraba con un tiempo materialmente limitado, sino con un calendario de posibilidades de actuación que no siempre era el más conveniente dentro de las anteriores limitaciones. Todo ello aconsejó decantarse por una metodología ágil y ligera, mediante la que poder desarrollar las tareas y objetivos de la mejor forma posible, teniendo en cuenta adicionalmente el condicionante económico en la etapa en que debía acometerse. De acuerdo con todo lo anterior, se determinó utilizar el estándar ESA como fórmula más adecuada para completar de forma satisfactoria el ciclo de vida del proyecto. Dicho estándar conlleva las siguientes fases:

- UR: Definición de requisitos de los usuarios.
- SR: Definición de requisitos del software.
- AD: Definición del diseño arquitectónico.
- DD: Diseño detallado y producción de código.
- TR: Transferencia del software a operaciones.
- OM: Operación y mantenimiento

A las que habrá que añadir dos fases más que serán la planificación del proyecto y su presupuesto.

6.2. Planificación del proyecto

La planificación llevada a cabo a lo largo del proyecto, considerando las tareas más importantes que se han realizado, se contiene en este apartado en el que se evalúa la planificación inicial y final, para posteriormente analizar las diferencias y desviaciones.

6.2.1. Planificación inicial

El desarrollo del proyecto se acomete en base a la constitución del entorno virtual y a la generación de los dos laboratorios: el de acceso web y el automático. Existen fases comunes para ambos laboratorios pero también existen fases independientes que se realizan para el desarrollo de uno de ellos pero que no son necesarias en el desarrollo del otro.

Se agrupa la planificación de forma global en diferentes fases, siendo las más importantes las siguientes:

- **Fase inicial**: En esta primera fase, el propósito consiste en asentar los diferentes ámbitos del proyecto, establecer una idea general del mismo y de su alcance y concretar una planificación inicial de las tareas a realizar.

  Se establecen unas 56 horas para realizar esta fase inicial.

- **Estudio y análisis general del entorno**: En esta segunda fase, se realiza una exploración completa de los diferentes hipervisores existentes en el mercado, un estado del arte de los mismos y la determinación de elegir uno de los sistemas para el desarrollo del proyecto. Además, se realizan unas pruebas mínimas en los hipervisores seleccionados para confirmar que éstos eran idóneos para los casos de uso.

  Se establecen una duración de unas 160 horas para completar esta fase.

- **Estudio de las herramientas de programación**: Se engloba en esta fase el aprendizaje de las nociones necesarias de programación tanto en Linux (bash) como en el lenguaje del hipervisor (Xen).

  Se planifican unas 56 horas para ejecutar esta fase.

- **Estudio de las herramientas de gestión**: Asimismo, se precisa un tiempo para realizar pruebas de instalación y configuración de las herramientas de gestión, con la finalidad de determinar la mejor elección para el laboratorio.

  La duración de esta fase se estableció en 112 horas.

- **Requisitos necesarios**: También es preciso establecer los requisitos necesarios para realizar el proyecto. Esta fase se centra más en las necesidades de configuración de los laboratorios, automático y web.

  Se previeron unas 48 horas para establecer los diferentes requisitos.

- **Diseño**: Una vez determinadas las diferentes herramientas, en esta fase se perfila el diseño de los laboratorios, definiendo su arquitectura y los principales elementos que lo van a formar.

  La realización del diseño se fijó en unas 150 horas.

- **Pruebas previas**: Se planifica una fase previa de pruebas para poder corroborar que el diseño está correctamente establecido y no existe ningún tipo de problema.

  Se establece una duración de 64 horas para acabar con esta fase.

- **Implementación final**: Esta fase abarca toda la implementación de los dos laboratorios. Es decir, abarca tanto la fase de implementación básica como la de implementación de valores añadidos.

  Se fijan unas 248 horas para la finalización de la fase de implementación.

- **Pruebas realizadas en los laboratorios**: También se considera necesario realizar una serie de pruebas para validar los laboratorios en un entorno productivo.

  Se establecen unas 60 horas para realizar las pruebas oportunas.

- **Documentación**: Aunque se realiza la documentación de forma paralela a la ejecución de las diferentes fases, se precisa de una última fase, adicional, para finalizar de maquetar la documentación.

  Se calcula un tiempo de unas 56 horas para terminar la última fase de la planificación.

A continuación se puede ver el diagrama de Gantt proyectado para la elaboración inicial del proyecto con un total de unas 1010 horas empleadas.

	Nombre de tarea	Duración	Comienzo	Fin	Predecesc
1	**PROYECTO FIN DE CARRERA**	**129 días**	**lun 08/10/12**	**jue 04/04/13**	
2	INICIO DEL PROYECTO	1 día	lun 08/10/12	lun 08/10/12	
3	FASE INICIAL	7 días	mar 09/10/12	mié 17/10/12	2
4	**ANÁLISIS**	**48 días**	**jue 18/10/12**	**lun 24/12/12**	
5	**Estudio y análisis general del entorno**	**20 días**	**jue 18/10/12**	**mié 14/11/12**	
6	Búsqueda documentación sobre hipervisores	10 días	jue 18/10/12	mié 31/10/12	3
7	Análisis de la documentación	14 días	mar 23/10/12	vie 09/11/12	6CC+3 días
8	Instalación hipervisores	8 días	mar 30/10/12	jue 08/11/12	7CC+5 días
9	Comparativa hipervisores	3 días	vie 09/11/12	mar 13/11/12	8
10	Elección hipervisor	1 día	mié 14/11/12	mié 14/11/12	9
11	**Estudio de las herramientas de programación**	**7 días**	**jue 15/11/12**	**vie 23/11/12**	
12	Programación bash	7 días	jue 15/11/12	vie 23/11/12	10
13	Programación xen	7 días	jue 15/11/12	vie 23/11/12	12CC
14	**Estudio de las herramientas de gestión**	**15 días**	**lun 26/11/12**	**vie 14/12/12**	
15	Estudio de OpenStack	7 días	lun 26/11/12	mar 04/12/12	13
16	Estudio de CloudStack	7 días	mié 05/12/12	jue 13/12/12	15
17	Elección herramienta de gestión	1 día	vie 14/12/12	vie 14/12/12	16
18	**Estudio resquisitos**	**6 días**	**lun 17/12/12**	**lun 24/12/12**	
19	Estudio requisitos usuario	6 días	lun 17/12/12	lun 24/12/12	17
20	Estudio requisitos software	6 días	lun 17/12/12	lun 24/12/12	19CC
21	Estudio requisitos hardware	6 días	lun 17/12/12	lun 24/12/12	19CC
22	**DISEÑO**	**19 días**	**mar 25/12/12**	**vie 18/01/13**	
23	Diseño arquitectura general	4 días	mar 25/12/12	vie 28/12/12	21
24	Diseño laboratorio web	10 días	lun 31/12/12	vie 11/01/13	23
25	Diseño laboratorio automático	5 días	lun 14/01/13	vie 18/01/13	24
26	**PRUEBAS PREVIAS**	**8 días**	**lun 21/01/13**	**mié 30/01/13**	
27	Realización pruebas web	4 días	lun 21/01/13	jue 24/01/13	25
28	Realización pruebas automáticas	4 días	vie 25/01/13	mié 30/01/13	27
29	**IMPLEMENTACION FINAL**	**31 días**	**jue 31/01/13**	**jue 14/03/13**	
30	Implementación general laboratorio	2 días	jue 31/01/13	vie 01/02/13	28
31	Implementación laboratorio web	24 días	lun 04/02/13	jue 07/03/13	30
32	Implementación laboratorio automático	25 días	vie 08/02/13	jue 14/03/13	30FC+4 días
33	**PRUEBAS Y EXPERIMENTOS**	**8 días**	**vie 15/03/13**	**mar 26/03/13**	
34	Pruebas entorno web	8 días	vie 15/03/13	mar 26/03/13	32
35	Ejecución experimentos malware	8 días	vie 15/03/13	mar 26/03/13	32
36	**DOCUMENTACION**	**7 días**	**mié 27/03/13**	**jue 04/04/13**	
37	Finalización documentación y maquetación	7 días	mié 27/03/13	jue 04/04/13	34;35

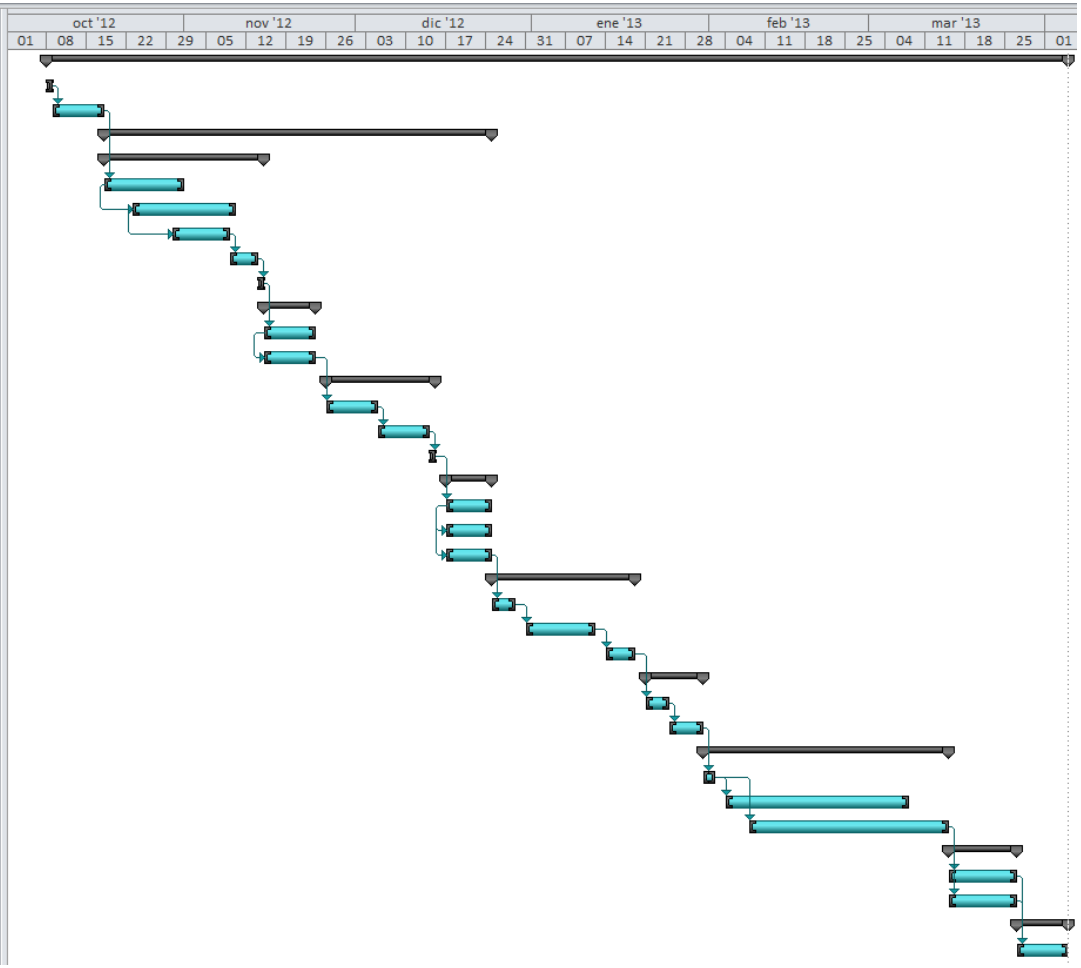


Figura 85. Diagrama Gantt planificación inicial.

6.2.2. Planificación real

Aunque la planificación inicial era bastante realista, su aplicación temporal, es decir, la utilización en el calendario de las horas previstas, experimentó algunas dificultades y contratiempos debido a un factor, el laboral, que condicionaba en gran manera la dedicación diaria a la realización del proyecto. En efecto, en función de las necesidades y decisiones de la empresa en la que presto mis servicios, hubo semanas en las que la dedicación era mayor que en otras, con independencia de la carga de trabajo que tuviese en cada momento. Este detalle variaba la dedicación diaria, lo que alargó la realización del proyecto en el tiempo. La duración final se estima en unos 7 meses.

6.3. Presupuesto

En cuanto al aspecto económico, este apartado contiene una estimación de los costes incurridos en la realización del proyecto. Los gastos se han clasificado en diferentes grupos, en función de su naturaleza: gastos de personal, gastos en equipamiento, gastos en bienes fungibles y, por último gastos en dietas y desplazamientos.

Para la realización de los diferentes presupuestos, se han tenido en cuenta las siguientes consideraciones:

- Los costes derivados de la realización del proyecto están expresados en Euros (€).
- Las cantidades que se presentan a continuación, se muestran con redondeo a las centésimas, es decir con dos decimales.

6.3.1. Presupuesto inicial

En base a la planificación inicial, se estimaron los siguientes costes asociados al proyecto con los que se elabora el presupuesto inicial.

6.3.1.1. Coste personal

En esta sección se detallan los costes incurridos por el personal que ha participado en el proyecto. Todos los gastos se estiman en función de los precios medios que aparecen en el régimen general de la seguridad social para el año 2012. [48]

Las bases de cotización para el grupo de cotización 1 (Ingenieros y Licenciados. Personal de alta dirección no incluido en el artículo 1.3.c) del Estatuto de los Trabajadores) son las siguientes:

[48] http://www.seg-social.es/Internet_1/Trabajadores/CotizacionRecaudaci10777/Basesytiposdecotiza36537/index.htm#36538

Grupo de Cotización	Categorías profesionales	Bases mínimas euros/mes	Bases máximas euros/mes
1	Ingenieros y Licenciados. Personal de alta dirección no incluido en el artículo 1.3.c) del estatuto de los trabajadores	1045,20	3262,50

Tabla 46. Bases de cotización contingencias comunes.48

Teniendo en cuenta una base de cotización media de 2217,30 €/mes y que la media de días trabajados al mes es de unos 20, se puede calcular un salario medio aproximado de 15 €/hora. Con estos datos, el coste del personal es el siguiente:

Nombre y Apellidos	Grupo de Cotización	€/hora	Horas Totales	Coste Total
Antonio Parra Truyol	1	15	1010	15.150€

Tabla 47. Coste bruto del personal.

Además del gasto bruto del personal, es preciso calcular el gasto en seguridad social del empleado, el gasto en formación profesional y el gasto por desempleo en función a las siguientes tablas:

Desempleo	Empresa	Trabajadores	Total
Tipo General	5,50	1,55	7,05
Contrato duración determinada tiempo completo	6,70	1,60	8,30
Contrato duración determinada tiempo parcial	7,70	1,60	9,30

Tabla 48. Bases de Cotización al desempleo. [49]

	Empresa	Trabajadores	Total
Formación profesional	0,60	0,10	0,70

Tabla 49. Coste en formación profesional.

Tipo Cotización %				
Contingencias comunes				Accidentes trabajo y Enfermedades profesionales
Grupo	**Empresa**	**Trabajador**	**Total**	
1	23,60	4,7	28,30	Tarifa primas disposición adicional cuarta Ley 42/2006, de 28 de diciembre, de Presupuestos Generales del Estado para 2007, según redacción de la disposición final octava de la Ley 26/2009, de 23 de diciembre, de Presupuestos Generales del Estado para 2010.
2 al 11	15,950	4,7	20,65	

Tabla 50. Tipo de cotización a la seguridad Social.

Los gastos incurridos derivados de la contratación de un empleado se pueden resumir en la siguiente tabla donde la base cotizada real se calcula en base a las horas reales

[49] http://www.seg-social.es/Internet_1/Trabajadores/CotizacionRecaudaci10777/Basesytiposdecotiza36537/index.htm#36538

estimadas que se haya trabajado al mes. En este caso, se puede estipular una media de 160 horas al mes. La base cotizada será de 160 horas * 15 €/hora = 2400 €/mes

Nombre y Apellidos	Base Cotizada €/mes	Tipo Cotización	Coste Total
Antonio Parra Truyol	2400	23,60 %	566,4€

Tabla 51. Coste en Seguridad Social del empleado.

Nombre y Apellidos	Base Cotizada €/mes	Tipo Cotización	Coste Total
Antonio Parra Truyol	2400	0,60 %	14,4€

Tabla 52. Coste en formación del empleado.

Nombre y Apellidos	Base Cotizada	Tipo Cotización	Coste Total
Antonio Parra Truyol	2400	7,70 %	184,8€

Tabla 53. Coste en prestación por desempleo del empleado.

Teniendo en cuenta todos los costes asociados al personal, y considerando una duración de 6 meses, el coste final que supondría el personal en la realización del proyecto sería el siguiente.

Concepto	Coste mensual (€)	Meses	Coste Total
Salario Bruto	2.400	6	14.400€
Cotización a la SS	566,4	6	3.398,4€
Cotización al desempleo	14,4	6	86,4€
Coste en formación	184,8	6	1.108,8€
Coste Total	3.165,6	6	**18.993,6€**

Tabla 54. Coste inicial total en personal.

6.3.1.2. Coste equipamiento

A continuación se detalla el coste imputable al proyecto del equipamiento hardware y de los recursos software utilizados en la realización del mismo.

Descripción	Precio Unitario	Cantidad	Coste Total	Amortización (meses)	Uso (meses)	Coste total imputable al proyecto
Portátil HP 6910p	1.600€	1	1.600€	48	6	200€
Microsoft Office Standard 2010	249€	1	249€	24	6	62,25€
Microsoft Visio 2010	330€	1	330€	24	1	13,75€
Microsoft Project 2010	1067€	1	1067€	24	1	44,46
Total						**320,46€**

Tabla 55. Coste inicial en equipamiento.

No se incluyen el *software* de licencia gratuita ya que no supone ningún coste aplicable al proyecto.

6.3.1.3. Coste bienes fungibles

Se estima un coste computable al proyecto para el material fungible (papel, tinta impresora…) de unos 70€

6.3.1.4. Costes indirectos

Se consideran costes indirectos al proyecto, el coste de la luz, de la conexión a internet, de los desplazamientos realizados…aplicables al coste total del proyecto. Estos costes son difíciles de cuantificar ya que presentan una alta volatilidad de precios. A continuación se establece una estimación de los costes indirectos, consistente en aplicar un coste por desplazamiento para reuniones o configuraciones in situ de 8€ con un total de unos 10 desplazamientos previstos mientras que, para el resto de conceptos, se han efectuado estimaciones en base al tiempo de utilización y tarifas de mercado. El conjunto de los costes indirectos estimados son:

Concepto	Coste mensual (€/mes)	Uso (meses)	Coste total imputable al proyecto
Conexión Internet	55	6	330€
Luz	15	6	90€
Desplazamientos			80€
Total			**500€**

Tabla 56. Costes iniciales indirectos.

6.3.1.5. Costes iniciales totales

En la siguiente tabla se exponen los costes totales presupuestados inicialmente. Se trata, obviamente, de la suma de los costes calculados anteriormente más el 21% de IVA.

Concepto	Coste sin IVA	Coste con IVA
Coste personal	18.993,6€	22.982,25€
Coste equipamiento	320,46€	387,75€
Coste bienes fungibles	70€	84,7€
Costes indirectos	500€	605€
Costes totales	**19.884,06€**	**24.059,7 €**

Tabla 57. Costes iniciales totales inicialmente presupuestados.

Por tanto, el coste final estimado para la realización del proyecto es de **24.059,7€**. Veamos a continuación el importe real que ha sido necesario para la llevar a cabo el proyecto.

6.3.2. Presupuesto final

Esta sección compara el presupuesto inicial con el coste efectivo necesario para realizar el proyecto. Las principales diferencias radican, como se ha apuntado, en las circunstancias laborales, que causaron que la planificación inicial y, por consiguiente, el presupuesto inicial, sufriesen una desviación considerable, si bien ésta es menos de cariz económico que de duración del proyecto.

Aunque al inicio del proyecto la disponibilidad estimada era de unos 6 meses en base a unas 5 horas diarias, la realidad fue muy diferente. Las 5 horas diarias no fueron tal, sino que fueron variables en función a la carga laboral de cada momento. Además, algunos fines de semana, por cuestiones asimismo laborales, no fue posible dedicar ni una hora al proyecto. Posteriormente, con las vacaciones de navidad y semana santa, pude dedicar más horas de las previstas al proyecto. Por último, tres semanas de vacaciones a tiempo completo dieron un empuje enorme a la finalización del mismo.

Como media, se determina que durante 4 meses la media de horas dedicadas al proyecto es de 5 horas, durante otros 4 la media es de 2 horas y durante 22 días, la media es de 12 horas diarias.

Esto hace un total de 1104 horas.

6.3.2.1. Coste personal

Teniendo en cuenta los datos obtenidos en el presupuesto inicial de coste en personal, y considerando una duración de 7 meses, el coste final que supone el personal en la realización del proyecto sería el siguiente.

Concepto	Coste mensual (€)	Meses	Coste Total
Salario Bruto	2.400	7	16.800€
Cotización a la SS	566,4	7	3.964,8€
Cotización al desempleo	14,4	7	100,8€
Coste en formación	184,8	7	1.293,6€
Coste Total	3.165,6	7	**22.159,2€**

Tabla 58. Coste final total en personal.

6.3.2.2. Coste equipamiento

Con los nuevos datos, el coste en equipamiento finalmente fue:

Descripción	Precio Unitario	Cantidad	Coste Total	Amortización (meses)	Uso (meses)	Coste total imputable al proyecto
Portátil HP 6910p	1.600€	1	1.600€	48	7	233,34€
Microsoft Office Standard 2010	249€	1	249€	24	7	72,63€
Microsoft Visio 2010	330€	1	330€	24	1	13,75€
Microsoft Project 2010	1067€	1	1067€	24	1	44,46
Total						**364,18€**

Tabla 59. Coste final en equipamiento.

6.3.2.3. Costes indirectos

Dentro de los costes indirectos, se aumentaron los desplazamientos de 10 a 18. Además, los costes totales se incrementaron al aumentar la duración del mismo.

Concepto	Coste mensual (€/mes)	Uso (meses)	Coste total imputable al proyecto
Conexión Internet	55	7	385€
Luz	15	7	105€
Desplazamientos			144€
Total			**634€**

Tabla 60. Costes finales indirectos.

6.3.2.4. Costes finales totales

En la siguiente tabla se establecen los costes totales presupuestados inicialmente. Se trata de la suma de los costes calculados anteriormente.

Concepto	Coste sin IVA	Coste con IVA
Coste personal	22.159,2€	26.812,63€
Coste equipamiento	364,18€	440,66€
Coste bienes fungibles	70€	84.7€
Costes indirectos	634€	767,14€
Costes totales	**23.227,38€**	**28.105,13€**

Tabla 61. Costes finales totales inicialmente presupuestados.

6.3.3. Análisis de la variación de costes

Haciendo un análisis de la variación de los presupuesto, se puede observar que inicialmente se presupuesta el proyecto con un coste de **24.059,7€** mientras que el presupuesto final es de **28.105,13€**. Esto supone una desviación de **4.045,43** lo que supone una variación del 16.8%.

Teniendo en cuenta que al presupuesto inicial del proyecto habría que sumarle un 20% de beneficio para asegurar la rentabilidad del proyecto más un 15% de margen de riesgo ate posibles adversidades, el presupuesto final hubiese sido de **32.480,6€.**

Esta cifra supone un beneficio final de **4.375,47€** siendo el beneficio final total de un 18.2%

6.4. Análisis del entorno tecnológico

En este apartado se analizan las herramientas hardware y software que se han utilizado para el desarrollo del entorno para el Grupo de Seguridad en las Tecnologías de la Información y las Comunicaciones. La solución, a pesar de haberse desarrollado en los servidores disponibles para el departamento, se podría haber llevado a cabo en cualquier equipo que estuviese equipado con tecnología de virtualización. De hecho, las pruebas iniciales para realizar las pruebas de concepto consistían únicamente de un portátil.

6.4.1. Herramientas Hardware

En este punto se realiza un estudio de las herramientas hardware utilizadas en la construcción del entorno virtual.

Equipo	Características
Servidor SuperMicro SERVICIOS	Procesador: Intel® Xeon® E5645 @2.40 GHz Número de Procesadores: 24 Memoria RAM: 12 GB NICs: 2 Disco Duro: 458 GB Coste: 1892,47€
Servidor SuperMicro PURPLE	Procesador: Intel® Xeon® E5645 @2.40 GHz Número de Procesadores: 24 Memoria RAM: 48 GB NICs: 2 Disco Duro: 458 GB Coste: 2430,86€
Servidor SuperMicro ENIGMA	Procesador: Intel® Xeon® E5645 @2.40 GHz Número de Procesadores: 24 Memoria RAM: 12 GB

Equipo	Características
	NICs: 2 Disco Duro: 458 GB Coste: 2430,86€
Servidor de Almacenamiento ABASTOR	Configuración: RAID 5 + Hot Spare disk Almacenamiento: 32 TB Servicios habilitados: ftp + nfs Memoria RAM: 8GB Canales DMA: 4 Coste: 8811,00€
Switch de 24 puertos	Coste: 536,76€
Portátil HP 6910p	Procesador: Intel® Core™2 Duo T7300 @2 GHz Memoria RAM: 4 GB Sistema Operativo: Windows 7 x86 NICs: 1 Disco Duro: 74,5 GB

Tabla 62. Herramientas Hardware.

6.4.2. Herramientas Software

En este punto se describen las herramientas *software* utilizadas en la realización del entorno virtual, así como una breve explicación de su funcionalidad.

Software	Funcionalidad
xe	Programa para el control, vía línea de comandos, de los servidores xen. Puede utilizarse en remoto y es gratuito.
Notepad ++	Editor de texto utilizado para la generación de los programas automatizados que generarán los laboratorios a través de línea de comandos utilizando xe. Gratuito.
CloudStack	Programa de gestión de nuestra nube. Con este programa se podrá gestionar la infraestructura de nuestra nube realizando las oportunas modificaciones. Gratuito
Microsoft Office 2010	Editor de texto utilizado para escribir el proyecto.
Google Drive	Servicio de almacenamiento y edición de documentos utilizado a lo largo de la realización del proyecto.
Microsoft Visio 2010	Aplicación para la creación de figuras y flujos de información.
Microsoft Proyect 2010	Aplicación para la planificación del proyecto.
Putty	Programa para la conexión remota a servidores.
WinSCP	Programa para la transferencia de ficheros entre equipos Linux y Windows.
Navegador web	Navegador web (IE, Chrome, Firefox…) con el que realizar la conexión al entorno web.

Tabla 63. Herramientas *software utilizadas.*

Capítulo 7

7. CONCLUSIONES Y LINEAS FUTURAS

Este capítulo resume las conclusiones a las que se han llegado en torno al proyecto así como las posibles líneas futuras que se podrían seguir para una eventual continuación del mismo.

7.1. Conclusiones

Un punto fundamental a destacar es que la evolución lógica de los sistemas no queda exenta del ataque del software malicioso o *malware*, por lo que su estudio, análisis y detección precoz constituyen los elementos más importantes para evitar que los sistemas estén desprotegidos e inseguros. En este proyecto se ha visto cómo poder detectar, en base a unas alertas específicas en los sistemas, si éstos han sido alterados de alguna manera o no. No todo el malware deja el mismo rastro pero sí tiene un patrón común en su proceso de infección.

Son precisamente estos rastros los que se precisa analizar para así anticiparse y minimizar el impacto que pudiesen ocasionar en los sistemas. Con ayuda del laboratorio automático, se pueden realizar pruebas aisladas donde sacar conclusiones y realizar mejoras en los sistemas. En concreto, el laboratorio ha sido diseñado para poder configurar de forma sencilla cada laboratorio, así como poder diseñar tantas baterías de pruebas como se desee.

La ejecución de los experimentos, de forma totalmente automática, ha permitido concluir que el laboratorio permite detectar comportamientos maliciosos en base a la presencia de eventos generados por las diferentes piezas de *malware*. Estos eventos variarán en función de la propia naturaleza del *malware.* Además, el aislamiento de los experimentos evita contaminaciones tanto del exterior como hacia el exterior y el automatismo permite la ejecución secuencial de experimentos sin necesidad de intervención externa. También se puede concluir que la gran cantidad de combinaciones de sistemas operativos y de aplicaciones permite crear entornos lo suficientemente flexibles como para crear una amplia base de datos de conocimiento de las diferentes piezas de *malware.*

Aunque los resultados obtenidos a lo largo de los experimentos, pueden aportar insuficiente información de forma aislada, si se procesa una gran cantidad de información y de toda esa cantidad de información se filtra la relevante, cruzándose y correlacionándose la misma, sí se puede obtener unos datos muy relevantes para detectar y evitar las vulnerabilidades en los sistemas.

En cualquier caso, siempre será posible efectuar un ajuste más fino para evitar falsos positivos o eventos que no aportan información útil a las pruebas. Además, al tratarse de una solución genérica, ésta puede adaptarse a las necesidades de cada usuario y, con la ayuda de la virtualización, distribuir los recursos físicos de los equipos para poder atender a las necesidades de cada momento. Realizar este tipo de experimentos en equipos físicos supondría un aumento del tiempo y del coste.

Por otro lado, el laboratorio web aporta un entorno amigable en el que poder desarrollar cualquier tipo de experimento. Se puede concluir que la naturaleza de este laboratorio permite a los usuarios acceder de forma sencilla y amigable a entornos aislados donde poder ejecutar pruebas aisladas y sencillas mediante *golpes* de ratón. Además, permite crear entornos colaborativos donde diferentes usuarios pueden interactuar en proyectos conjuntos donde cada uno configura su pieza para finalmente crear una red donde ejecutar las pruebas.

7.2. Líneas Futuras

Como líneas futuras, se proponen diferentes posibilidades para poder continuar con el trabajo realizado en este proyecto. Estas posibilidades pretenden aportar mejoras y un valor añadido al mismo.

En relación al laboratorio automático se proponen las siguientes ideas:

- Minería de datos. En base a este proyecto se pueden desarrollar programas de cómputo inteligentes, capaces de utilizar la información para tomar una serie de medidas.
- Proxy inteligente para dar salida a internet. Se ha comentado en el apartado de Experimentos el hecho de que los ejecutables intenten realizar una conexión a Internet para efectuar la descarga de ficheros. Pues bien, con un proxy inteligente se podría ampliar la cantidad de malware a evaluar. Además, el servidor de OSSIM podría realizar actualizaciones de ficheros o configuraciones en línea.
- Optimizar el escaneo de los sistemas. Es posible mejorar la cantidad de directorios y claves de registro que se utilizan para monitorizar el sistema con el objetivo de minimizar el tiempo de escaneo del mismo sin comprometer la seguridad.
- Crear nuevas alertas, o excluir alguna para mejorar el entorno. Actualmente se alerta de una serie de eventos, pero es posible modificar las alertas para detectar eventos no monitorizados o excluir algún evento que resulte trivial.

- Realizar una correlación de los datos obtenidos. De este modo podría obtenerse una visión más general de los eventos.
- Utilizar distintos sistemas operativos Windows y ampliar el análisis a sistemas Linux. La evolución de los sistemas es continua, por lo que puede resultar interesante incluir nuevos sistemas operativos Windows y ampliar el análisis a sistemas Linux.
- Reducir el tiempo de ejecución del experimento mediante optimizaciones en los sistemas operativos y en la instalación de las aplicaciones.
- Ejecución secuencial de experimentos con el mismo *malware* para determinar indicios de vulnerabilidades en los sistemas.

Y del laboratorio web, se presentan las siguientes:

- Utilizar diferentes hipervisores. Ampliar la variedad de hipervisores para determinar la eficiencia de cada uno.
- Crear nubes privadas con otras herramientas disponibles. Se podría efectuar una evaluación y comparativa de las mismas.
- Creación de un entorno de teletrabajo. Es posible diseñar una nube en la que los usuarios puedan conectarse desde Internet y disponer de un escritorio propio con sus herramientas. Para ello habría que realizar un estudio de las reglas de cortafuegos que permita securizar el entorno.

Glosario, términos y acrónimos.

TIC	Tecnologías de la Información y la Comunicación
COSEC	Grupo de Seguridad en las Tecnologías de la Información y las Comunicaciones
Malware	Software malicioso cuyo fin principal es infiltrarse en un sistema aprovechándose de cualquier vulnerabilidad que presente el sistema,
Hipervisor	Capa intermedia entre el hardware del equipo donde se instala y los diferentes entornos virtuales que pueden correr sobre él.
HW	Hardware
CPD	Centro de Procesado de Dtos
RAM	Random-Access Memory
XCP	Xen Cloud Platform
KVM	Kernel-based Virtual Machine
Cloud Computing	Tecnología que permite ofertar servicios de cómputo y almacenamiento de forma remota, a través de una red.
SaaS	Software as a Service
PaaS	Platform as a Service
IaaS	Infraestructure as a Service
API	Application Programming Interface
Ip	Internet Protocol
NFS	Network File System
iSCSI	Internet Small Computer System Interface
vhd	Virtual hard disk

ISO	Tipo de fichero de un disco óptico
VLAN	Virtual Local Area Network
NAT	Network Address Translation
OSSIM	Open Source Security Information Management
SIEM	Security Information and Event Management
SIM	Security Information Management
SEM	Security Event Manager
HIDS	Host-Based Intrusion Detection System
OSSEC	Open source HIDS
LMS	Log Management System
SLM/SEM	Security Log/Event Management
SIM	Security Information Management
SEC	Security Event Correlation
CPU	Central Processing Unit
SMTP	Simple Mail Transfer Protocol
DHCP	Dynamic Host Configuration Protocol
NIC	Network Interface Controller
MSDNAA	Microsoft Developer Network Academic Alliance
NAS	Network-attached storage
http	Hypertext Transfer Protocol
https	Hypertext Transfer Protocol Secure

BIBLIOGRAFIA

[1] Stephen Soltesz, Princeton University, Herbert Potzl, Linux-VServer Maintainer, Marc E. Fiuczynski, Princeton University, Andy Bavier, Princeton University, Larry Peterson, Princeton University. "*Container-based Operating System Virtualization.*" Proceedings of the 2nd ACM SIGOPS/EuroSys European Conference on Computer Systems 2007, Marzo 2007. [Consulta: Octubre 2012]

[2] Steven Hand, Tim Harris, Evangelos Kotsovinos, Ian Pratt, University of Cambridge Computer Laboratory, J J Thomson Avenue, Cambridge, UK, CB3 0FD. "*Controlling the XenoServer Open Platform.*" [Consulta: Octubre 2012]

[3] David Chisnall, "The Definitive Guide to the Xen Hypervisor", Prentice Hall, Noviembre 2007. [Consulta Octubre 2012]

[4] Historia de Xen. Universidad de Cambridge. [Internet]: http://www.xen.org/community/xenhistory.html [Consulta: Octubre 2012]

[5] Historia de Xen. Wikipedia. http://en.wikipedia.org/wiki/Xen [Consulta: Octubre 2012]

[6] Gabe Knuth. "*A brief history of Xen and XenSource.*" 16 Agosto 2007. [Internet]: http://www.brianmadden.com/blogs/gabeknuth/archive/2007/08/16/a-brief-history-of-xen-and-xensource.aspx [Consulta: Octubre 2012]

[7] VMWare ® "*Understanding Full Virtualization, Paravirtualization, and Hardware Assist*" Septiembre 2007. [Consulta: Octubre 2012]

[8] Miguel Vidal, José Castro "*The art of virtualization with free software*" Abril 2010. [Consulta: Octubre 2012]

[9] Alberto Abián Belmonte, UCM "*Virtualización en GNU/Linux*" WhyFloss Conference, Madrid, Julio 2007. [Consulta: Octubre 2012]

Bibliografía Relativa a la Virtualización

[10] Definición de virtualización. Wikipedia [Internet]: http://en.wikipedia.org/wiki/Virtualization [Consulta: Octubre 2012]

[11] Definición de virtualización y su diferente clasificación Red Hat. [Internet]: https://access.redhat.com/knowledge/docs/en-US/Red_Hat_Enterprise_Linux/5/html/Virtualization/pr01s05.html [Consulta: Octubre 2012]

[12] Definición de virtualización y su diferente clasificación. [Internet]: http://decipherinfosys.wordpress.com/2009/02/12/virtualization-basics-and-the-different-types-of-virtualization/ [Consulta: Octubre 2012]

[13] Definición de virtualización y su diferente clasificación Datamation. [Internet]: http://www.datamation.com/netsys/article.php/3884091/Virtualization.htm [Consulta: Octubre 2012]

Bibliografía Relativa a los Hipervisores

[14] Comparativa de hipervisores de tipo 1 y 2.Datamation®. [Internet]: http://virtualizationreview.com/blogs/everyday-virtualization/2009/06/type-1-and-type-2-hypervisors-explained.aspx [Consulta: Noviembre 2012]
[15] Definición de hipervisor. Wikipedia [Internet]: http://en.wikipedia.org/wiki/Hypervisor [Consulta: Octubre 2012]
[16] Hipervisores. Laboratory for Advanced System Software. [Internet]: http://lass.cs.umass.edu/~shenoy/courses/677/lectures/Lec05.pdf [Consulta: Octubre 2012]
[17] Diferentes tipos de hipervisores. Virtual Computer ™. [Internet]: http://www.virtualcomputer.com/type-1-vs-type-2-hypervisor [Consulta: Octubre 2012]
[18] Diferencias entre Hipervisores tipo 1 y tipo 2. Citrixcdn. [Internet]: http://www.youtube.com/watch?v=1YtRukUeg04 [Consulta: Octubre 2012]
[19] Informe de riesgos y tendencias. IBM®. [Internet]: ftp://public.dhe.ibm.com/common/ssi/ecm/en/wgl03003usen/WGL03003USEN.PDF [Consulta: Octubre 2012]
[20] Visión de los sistemas virtuales. IBM®. [Internet]: http://publib.boulder.ibm.com/infocenter/eserver/v1r2/index.jsp?topic=%2Feicay%2Feicayvservers.htm [Consulta: Octubre 2012]
[21] Hipervisores y sus propiedades. Wikipedia. [Internet]: http://en.wikipedia.org/wiki/Embedded_hypervisor [Consulta: Octubre 2012]
[22] Descripción de VMWare. Wikipedia. [Internet]: http://en.wikipedia.org/wiki/VMware [Consulta: Octubre 2012]
[23] Arquitectura de VMWare. VMWare®. [Internet]: http://www.vmware.com/pdf/vi_architecture_wp.pdf [Consulta: Octubre 2012]

Comparativa Xen, VMware y KVM

[24] Descripción de Xen y comparativa con otros hipervisores. XEN. [Internet]: http://www.xen.org/files/Marketing/WhyXen.pdf [Consulta: Noviembre 2012]
[25] Comparativa de hipervisores. Datamation®. [Internet]: http://www.datamation.com/feature/Virtual-Server-Comparison-Xen-vs-Microsoft-vs-VMware-2010-3853716-3.htm [Consulta: Noviembre 2012]
[26] Cuadrante mágico. Tecnologías de virtualización. Gartner®. [Internet]: http://www.gartner.com/technology/reprints.do?id=1-1B2IRYF&ct=120626&st=sg

Cloud Computing

[27] Definición de *cloud computing*. Wikipedia. [Internet]: http://en.wikipedia.org/wiki/Cloud_computing [Consulta: Noviembre 2012]
[28] Definición de *cloud computing*. Eric Knorr y Galen Gruman. InfoWorld. [Internet]: http://www.infoworld.com/d/cloud-computing/what-cloud-computing-really-means-031 [Consulta: Noviembre 2012]

[29] Definición de *cloud computing*. IBM®. [Internet]: http://www.ibm.com/cloud-computing/us/en/what-is-cloud-computing.html [Consulta: Noviembre 2012]
[30] Seguridad en entornos de *cloud computing*. Wikipedia. [Internet]: http://en.wikipedia.org/wiki/Cloud_computing_security [Consulta: Noviembre 2012]
[31] Definición de *cloud computing*. Citrix®. [Internet]: http://www.youtube.com/watch?v=I8RwggLRlN8 [Consulta: Noviembre 2012]
[32] Pros y contras de cloud computing. Kristi Holland. [Internet]: http://www.thebeckon.com/pros-and-cons-of-cloud-computing/ [Consulta: Noviembre 2012]
[33] Nubes híbridas. [Internet]: http://www.djeek.com/2011/02/hybridcloud/ [Consulta: Noviembre 2012]

Comparativa OpenStack CloudStack

[34] Comparativa OpenStack y CloudStack. Sam Su. [Internet]: http://cloud-vaporware.blogspot.com.es/2011/11/cloudstack.html [Consulta: Diciembre 2012]
[35] "OpenStack vs. CloudStack: The beginning of the open-source cloud wars". Steven J. Vaughan-Nichols. [Internet]: http://www.zdnet.com/blog/open-source/openstack-vs-cloudstack-the-beginning-of-the-open-source-cloud-wars/10763 [Consulta: Diciembre 2012]
[36] "Citrix, CloudStack, OpenStack, and the war for open-source clouds". Lydia Leong. [Internet]: http://blogs.gartner.com/lydia_leong/2012/04/03/citrix-cloudstack-openstack-and-the-war-for-open-source-clouds/ [Consulta: Diciembre 2012]
[37] "OpenStack versus CloudStack: A contest between services and software". David Linthicum. [Internet]: http://www.infoworld.com/d/cloud-computing/openstack-versus-cloudstack-contest-between-services-and-software-190225 [Consulta: Diciembre 2012]
[38] "Cloud platform comparison: CloudStack, Eucalyptus, vCloud Director and OpenStack". Vadim Truksha. [Internet]: http://www.networkworld.com/news/tech/2012/071612-cloud-platform-comparison-260923.html [Consulta: Diciembre 2012]
[39] "CloudStack vs. OpenStack: Smackdown On, Who Wins?" Mike Barton. [Internet]: http://www.wired.com/insights/2012/06/cloudstack-vs-openstack/ [Consulta: Diciembre 2012]
[40] "The Stack Wars: OpenStack vs. CloudStack". Joe Onisick. [Internet]: http://www.networkcomputing.com/cloud-computing/the-stack-wars-openstack-vs-cloudstack/240000933 [Consulta: Diciembre 2012]

OSSIM

[41] Definición de SIEM. Wikipedia. [Internet]: http://en.wikipedia.org/wiki/Security_information_and_event_management [Consulta: Diciembre 2012]

[42] Descripción del producto. Alienvault. OSSIM.[Internet]: http://communities.alienvault.com/?utm_expid=61134069-1 [Consulta: Diciembre 2012]
[43] Definición de OSSIM. Wikipedia. [Internet]: http://en.wikipedia.org/wiki/OSSIM [Consulta: Diciembre 2012]
[44] Foro sobre OSSIM. Alienvault. [Internet]: https://forums.alienvault.com [Consulta: Diciembre 2012]
[45] Información acerca del agente de OSSEC. [Internet]: http://www.ossec.net/ [Consulta: Diciembre 2012]
[46] Definición de OSSEC. Wikipedia. [Internet]: http://en.wikipedia.org/wiki/OSSEC [Consulta: Diciembre 2012]
[47] Definición de sistemas HIDS. Wikipedia. [Internet]: http://en.wikipedia.org/wiki/Host-based_intrusion_detection_system [Consulta: Diciembre 2012]

Programas de ayuda

[48] DenyHosts. [Internet]: http://denyhosts.sourceforge.net/ [Consulta: Enero 2013]
[49] Generación de clavespúblicas y privadas en Sistemas Ubuntu. http://lani78.wordpress.com/2008/08/08/generate-a-ssh-key-and-disable-password-authentication-on-ubuntu-server/ [Consulta: Enero 2013]
[50] Iptable. Nixcraft. [Internet]: http://www.cyberciti.biz/tips/linux-iptables-9-allow-icmp-ping.html [Consulta: Enero 2013]
[51] Redirección del tráfico http a https. [Internet]: http://irisesupport.onconfluence.com/display/kbase/Redirecting+HTTP+to+HTTPS+in+Tomcat [Consulta: Enero 2013]
[52] Configuración de NAT en Ubuntu. Foro Ubuntu. [Internet]: http://ubuntuforums.org/showthread.php?t=1715735 [Consulta: Enero 2013] https://help.ubuntu.com/community/Router#Enable_IP_forwarding_and_Masquerading [Consulta: Enero 2013] http://linux.about.com/od/ubusrv_doc/a/ubusg18t03.htm [Consulta: Enero 2013]
[53] "Sysprep a Windows 7 Machine – Start to Finish". Brian Jackson. [Internet]: http://theitbros.com/sysprep-a-windows-7-machine-start-to-finish-v2/ [Consulta: Febrero 2013]
[54] "Cómo utilizar la herramienta Sysprep para automatizar la correcta implementación de Windows XP". Microsoft®. [Internet]: http://support.microsoft.com/kb/302577/es [Consulta: Febrero 2013]

CloudStack

[55] Definición de SIEM. Wikipedia. [Internet]: http://en.wikipedia.org/wiki/Cloudstack [Consulta: Noviembre 2012]
[56] Documentación CloudStack. CloudStack. [Internet]: http://www.cloudstack.org/ [Consulta: Noviembre 2012]

[57] Instalación de CloudStack. CloudStack. [Internet]:

[58] http://docs.cloudstack.org/CloudStack_Documentation/Installing_CloudStack [Consulta: Febrero 2013]

[59] Confiuración de Mysql. MySql. [Internet]: http://dev.mysql.com/ [Consulta: Febrero 2013]

[60] "Communication ports". Citrix® [Internet]: http://support.citrix.com/servlet/KbServlet/download/2389-102-654859/CitrixPorts_by_Port_1103.pdf [Consulta: Febrero 2013]

[61] Configuración avanzada de Zonas en Cloudstack. Mayur Dhande [Internet]: http://blogs.clogeny.com/citrixs-cloudstack-3-0-advanced-zone-setup/ [Consulta: Febrero 2013]

[62] Configuración de https en CloudStack. CloudStack. [Internet]: http://docs.cloudstack.org/Knowledge_Base/Enable_HTTPS_for_CloudStack_Web_Interface [Consulta: Febrero 2013]

OpenStack

[63] Definición de SIEM. Wikipedia. [Internet]: http://en.wikipedia.org/wiki/OpenStack [Consulta: Noviembre 2012]

[64] "Introducción a OpenStack" Belmiro Daniel Rodrigues Moreira [Internet]: http://indico.cern.ch/getFile.py/access?contribId=0&resId=0&materialId=slides&confId=130854 [Consulta: Noviembre 2012]

[65] Documentación de OpenStack. OpenStack. [Internet]: http://www.openstack.org/ [Consulta: Noviembre 2012]

OpenXenManager

[66] Configurar OpenXenManager en Sistema Ubuntu. Souceforge. [Internet]: http://sourceforge.net/apps/trac/openxenmanager/wiki/GettingStarted#Ubuntu/Debian [Consulta: Febrero 2013]

[67] OpenXenManager alternative to XenCenter. [Internet]: http://www.dedoimedo.com/computers/openxenmanager.html [Consulta: Febrero 2013]